AD-A052 679 B-K DYNAMICS INC ROCKVILLE MD F/G 9/2
QUARTERLY INTERIM TECHNICAL REPORT (3RD), CONTRACT DAAH01-75-C---ETC(U)
JUL 75 DAAH01-75-C-0194
UNCLASSIFIED BKD-TR-3-197 NL

1 OF
AD
AD52679

(1)

(14) BKD-TR-3-197

(6) Quarterly Interim Technical Report (3rd)
Contract DAAHØ1-75-C-Ø194

(11) 22 Jul 75 (12) 179 p.

(15) DAAHØ1-75-C-Ø194

(9) Rept. for 15 Apr – 15 Jul 75.

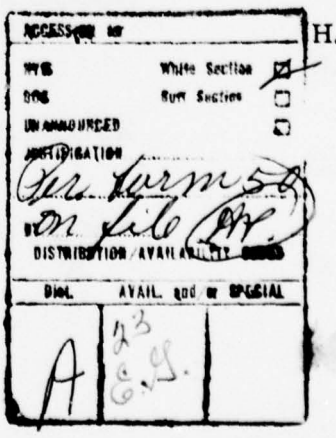
BOK
DYNAMICS, INC.

387 575

TABLE OF CONTENTS

B&K
DYNAMICS, INC.

TABLE OF CONTENTS



LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

B-K Dynamics' activities during the third quarter (15 April 1975 - 15 July 1975) continued to focus on preparation of hardware for the interim STINGER simulation and conversion of the STINGER simulation to the ASC's hybrid computer system. This report describes the STINGER conversion activities and presents program flow charts and coding generated in this task.

2.0 STINGER CONVERSION TO ASC EQUIPMENT

Converting the STINGER real-time simulation from the interim system to the ASC equipment has required replacing IBM-7094 software functions with equivalent CDC-6600 functions, and IBM-DOS interface operations with equivalent DADIOS ADC, DAC and discrete word handling capabilities. In Figure 2.1 the new ASC equipment utilized in phase 1 conversion is shown in relation to the overall STINGER simulation. The items which represent new ASC equipment are enclosed in dotted lines.

The documentation given herein, of the STINGER simulation, reflects the rationale and realization of an attempt to duplicate interim STINGER functions on ASC equipment. This conversion took place within the constraints of available documentation, installation of new hardware, availability of new hardware, and an attempt to minimize the effect of phase 1 conversion on other modules of the simulation. The phase 1 STINGER conversion presented in the following sections represents a one-on-one conversion of software and hardware.

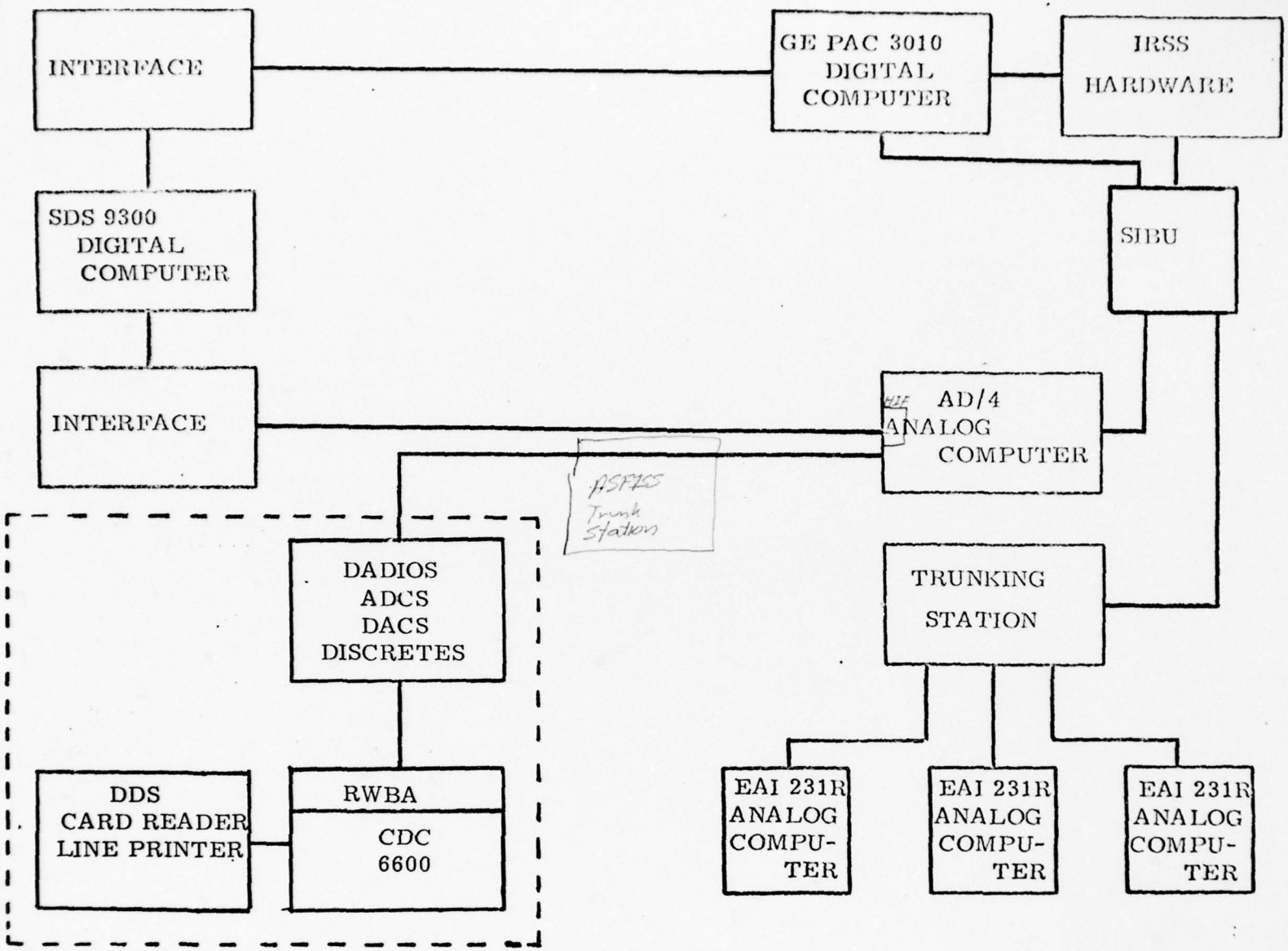


FIGURE 2.1 PHASE 1 CONVERSION OF STINGER SIMULATION

2.1 SOFTWARE CONVERSION OF REAL-TIME CODE

Previously, the portion of the interim STINGER simulation implemented on the IBM/7094 consisted of approximately 2000 lines of code in FORTRAN and 500 lines of code in MAP (IBM-7094 assembly language). Converting this software to the CDC-6600 required minor changes to the FORTRAN program* and completely new code written in FORTRAN hybrid to replace the IBM-7094 assembly code.

This new code performs the identical functions of the interim simulation. These functions include pre-real-time data processing, real-time computations and post-real-time data analysis. A summary of the functions performed by each of these three major program segments is summarized in Appendix G.

In Figure 2.2, a flow chart of the real-time portion of the phase 1 STINGER program is presented. An explanation of the symbols used in Figure 2.2 are given in Table 2.1. Furthermore, a limited description of the computational equations are given in Appendix A. These equations represent a one-on-one cross reference between program symbols and mathematical notation.

The real-time software presented here has been tested in real-time using simplified test functions. The simplified test functions used are presented as part of a FORTRAN hybrid real-time subroutine in Appendix H. These simplified test functions have been used in the absence of real test data from the existing simulation.

*The modifications to the FORTRAN program were performed by MICOM personnel.

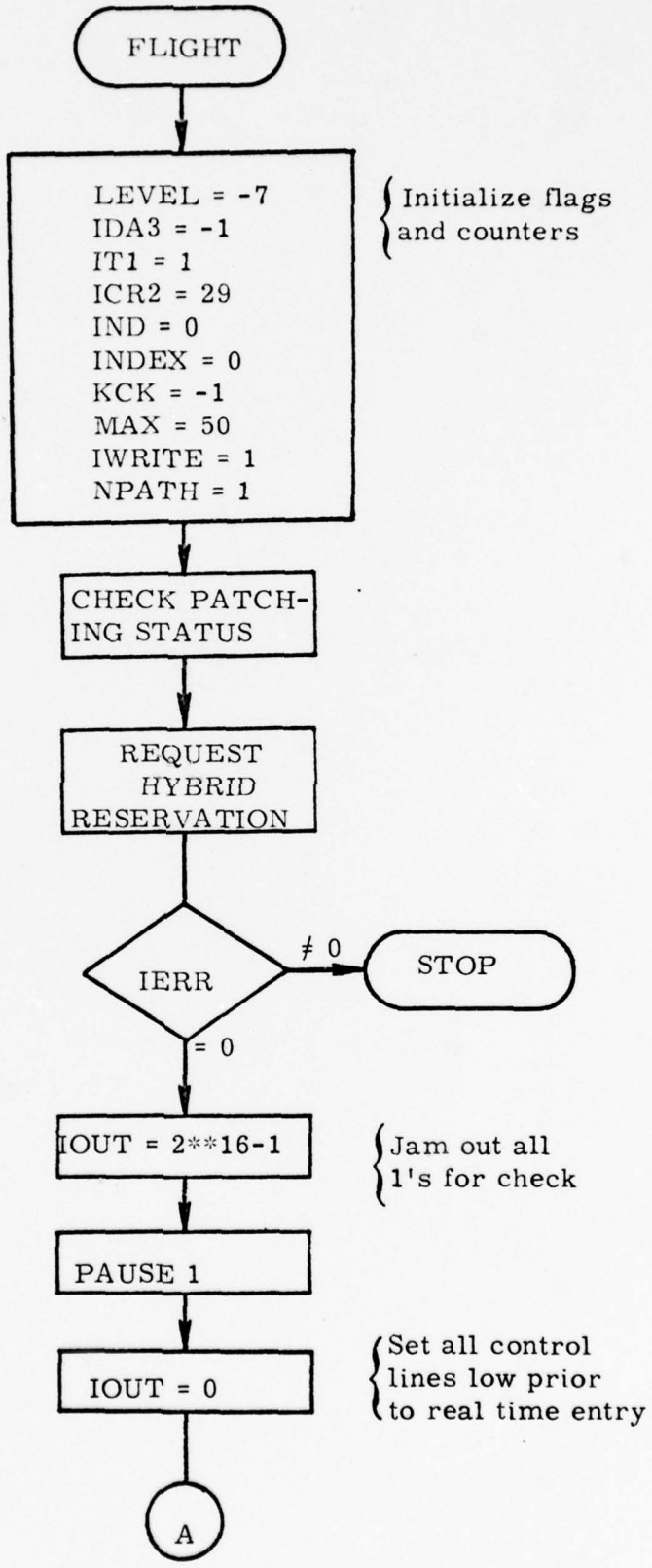
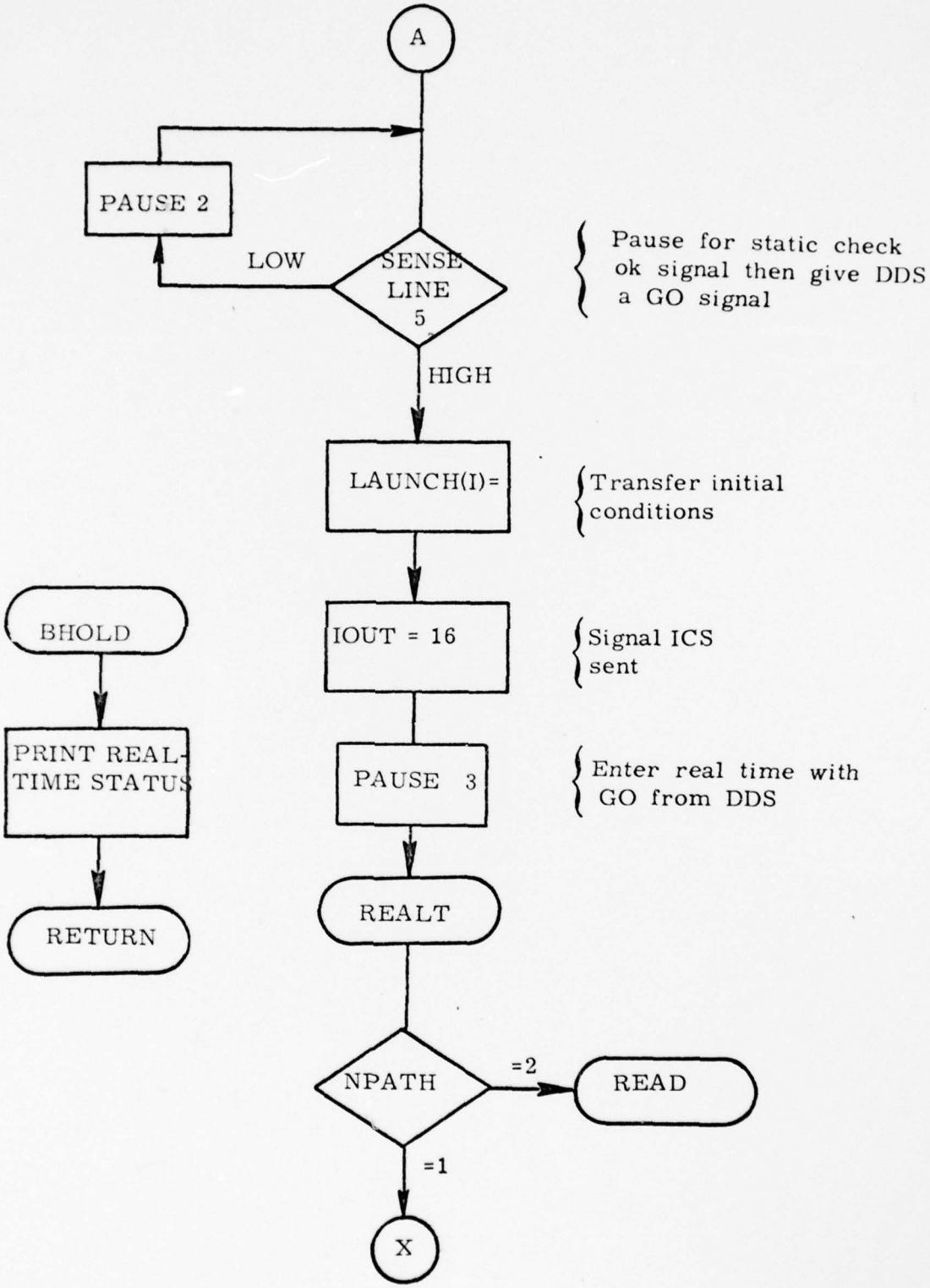
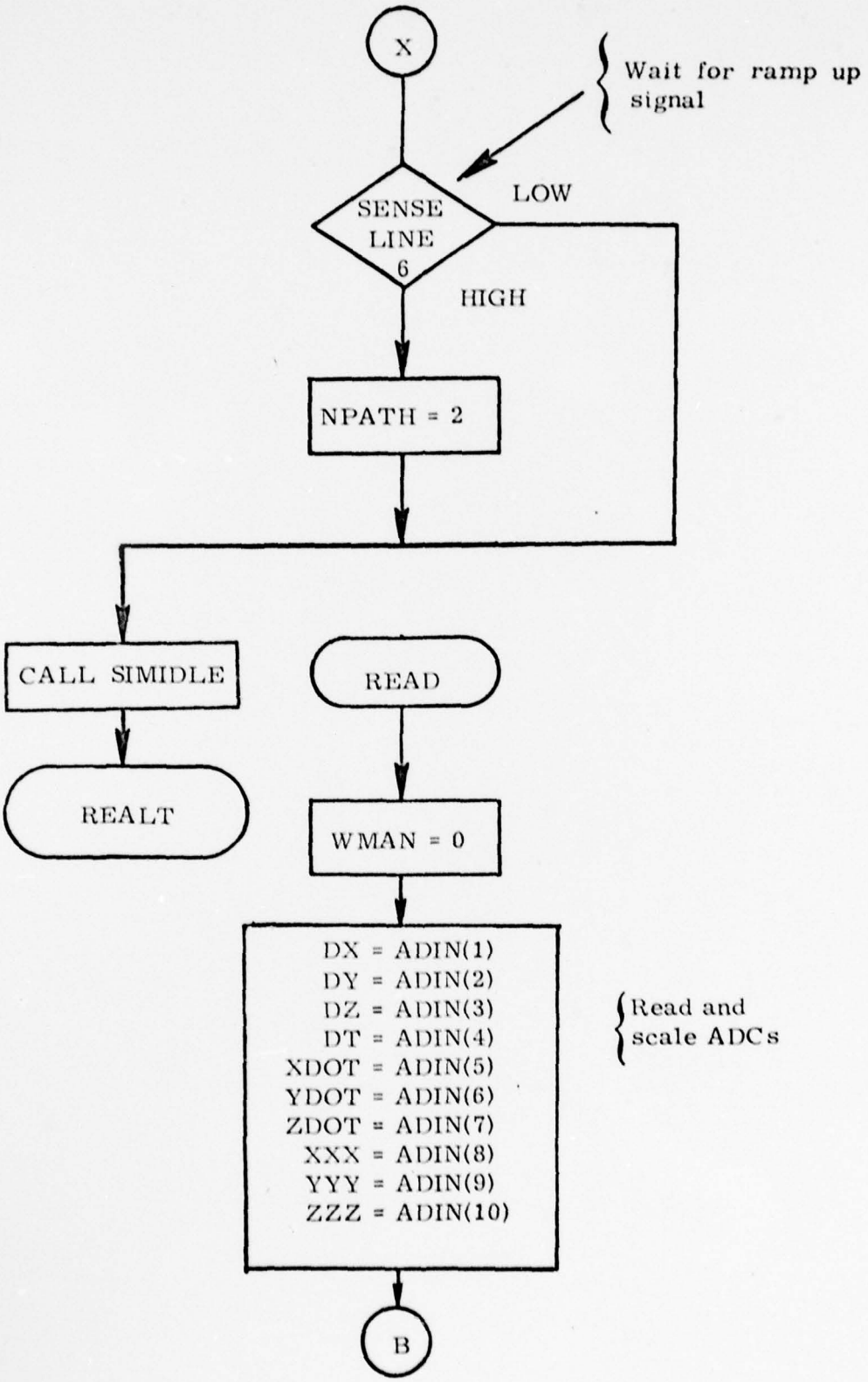
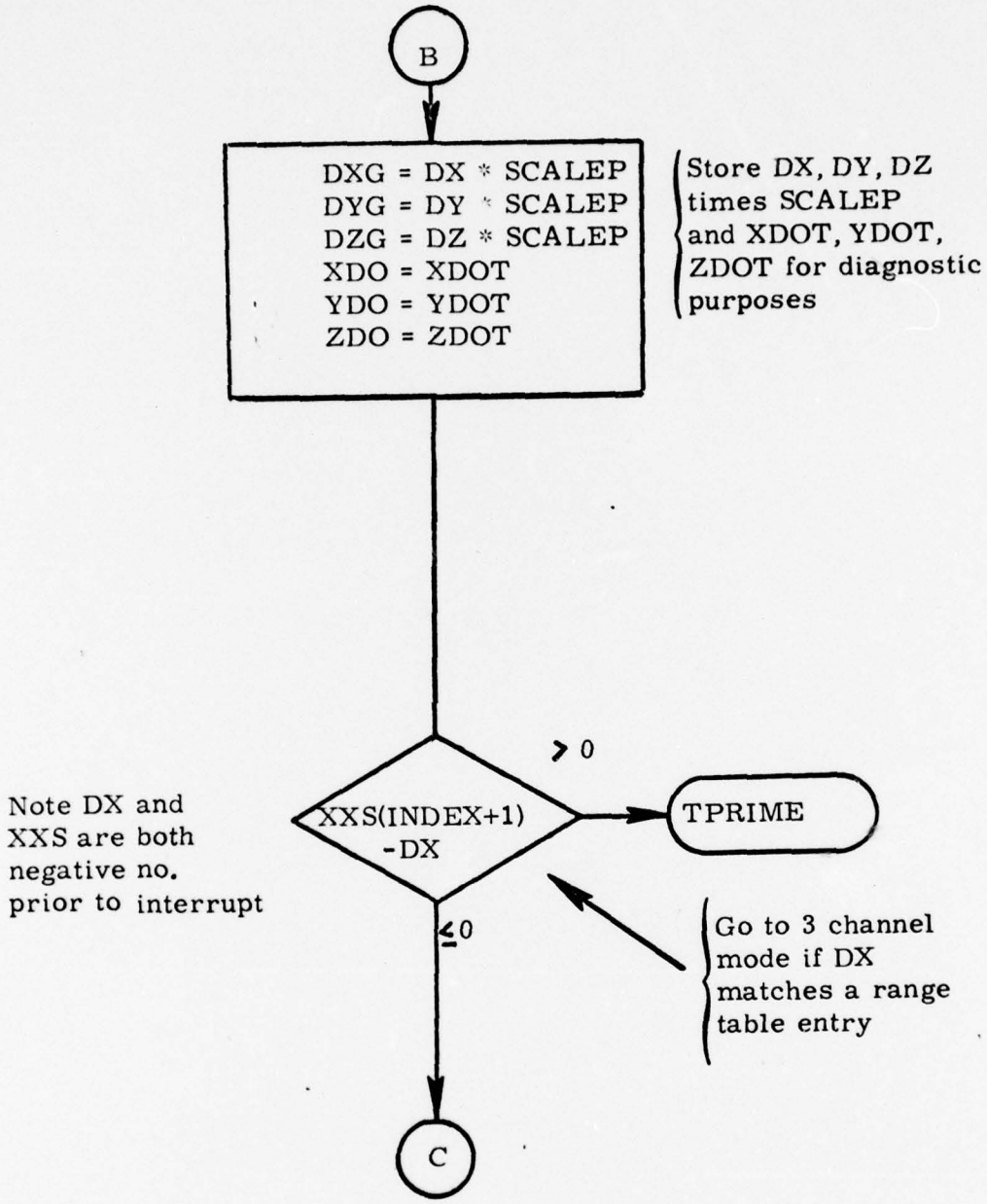
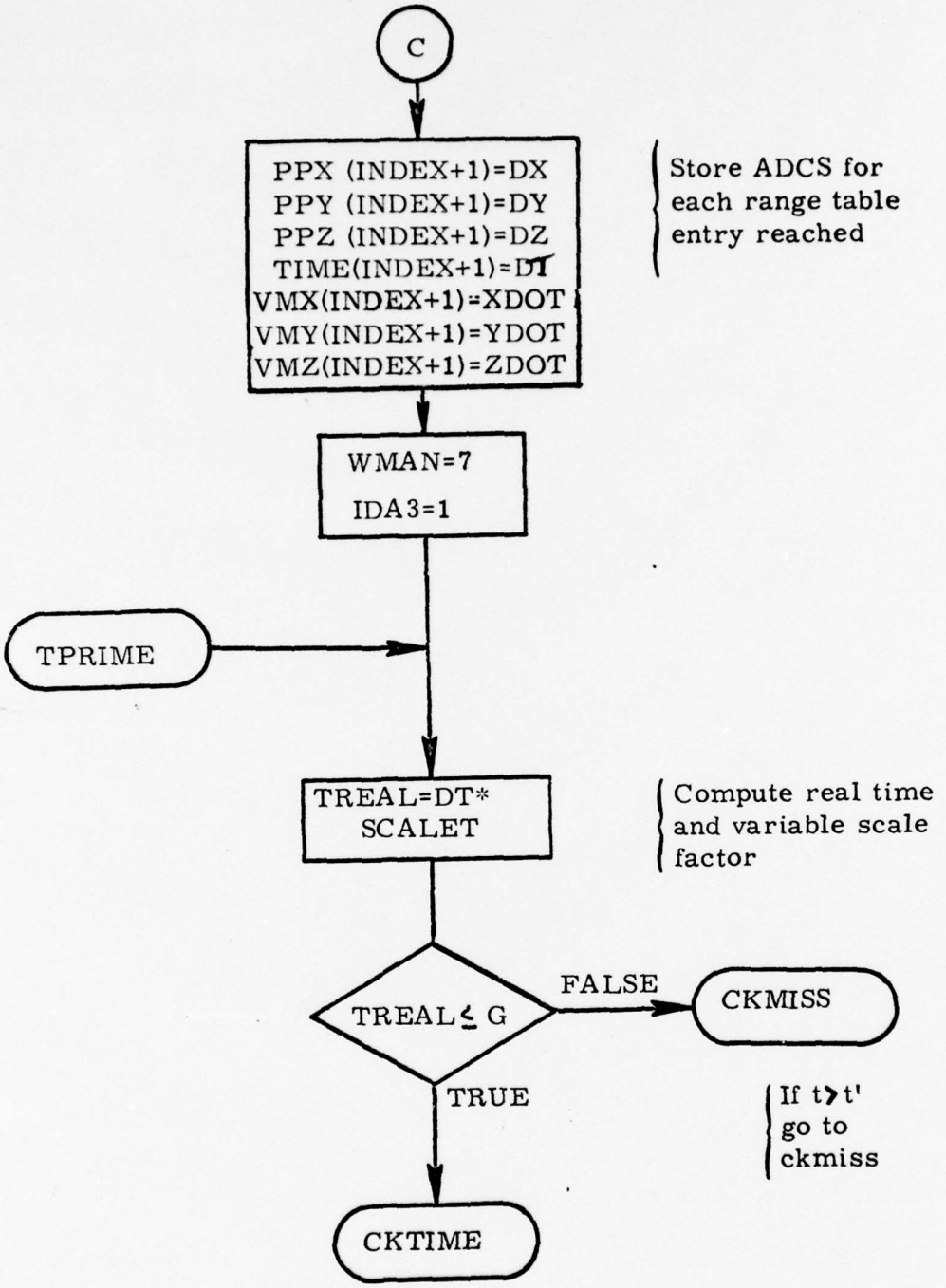


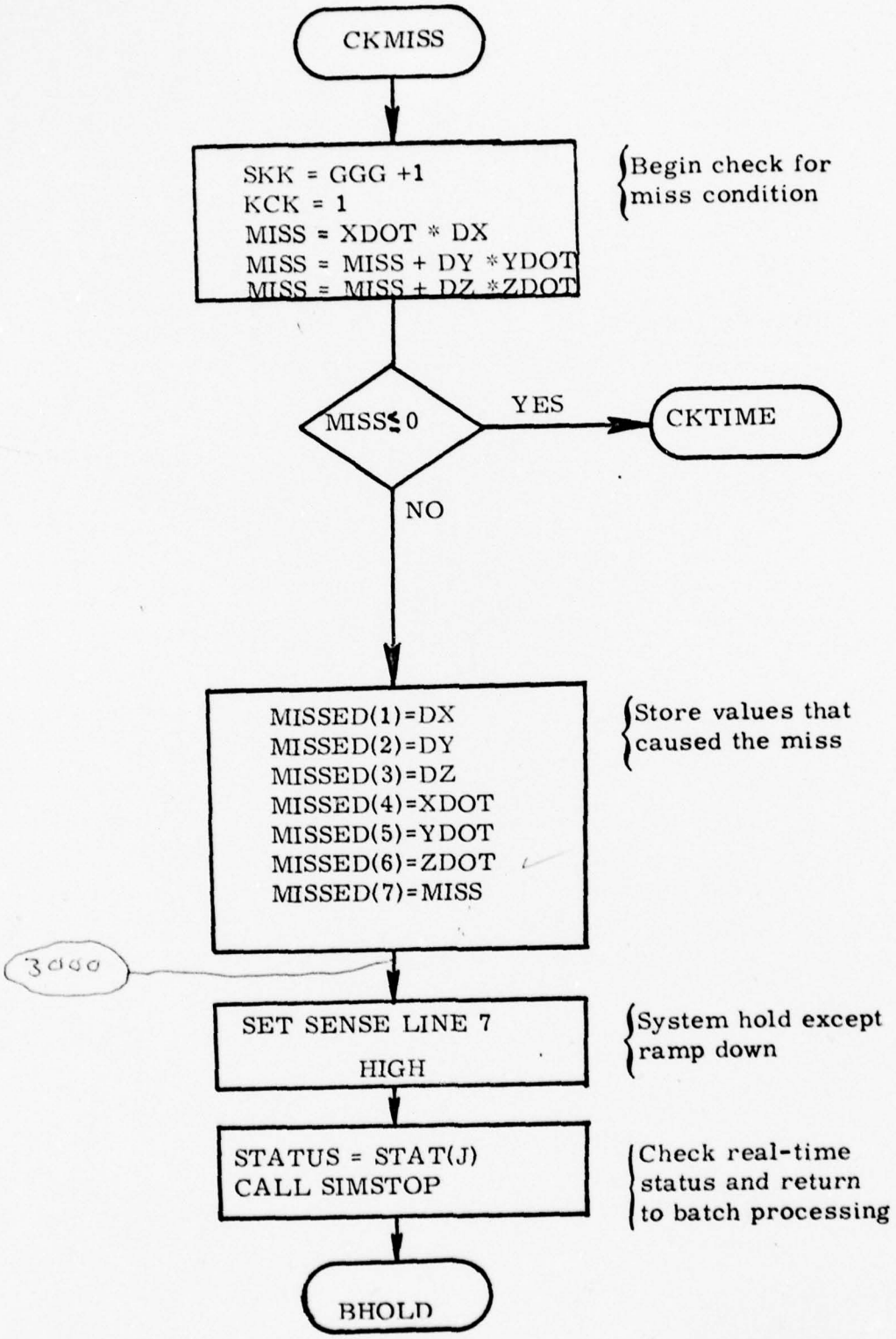
FIGURE 2.2 FLOW CHART OF STINGER REAL TIME CODE

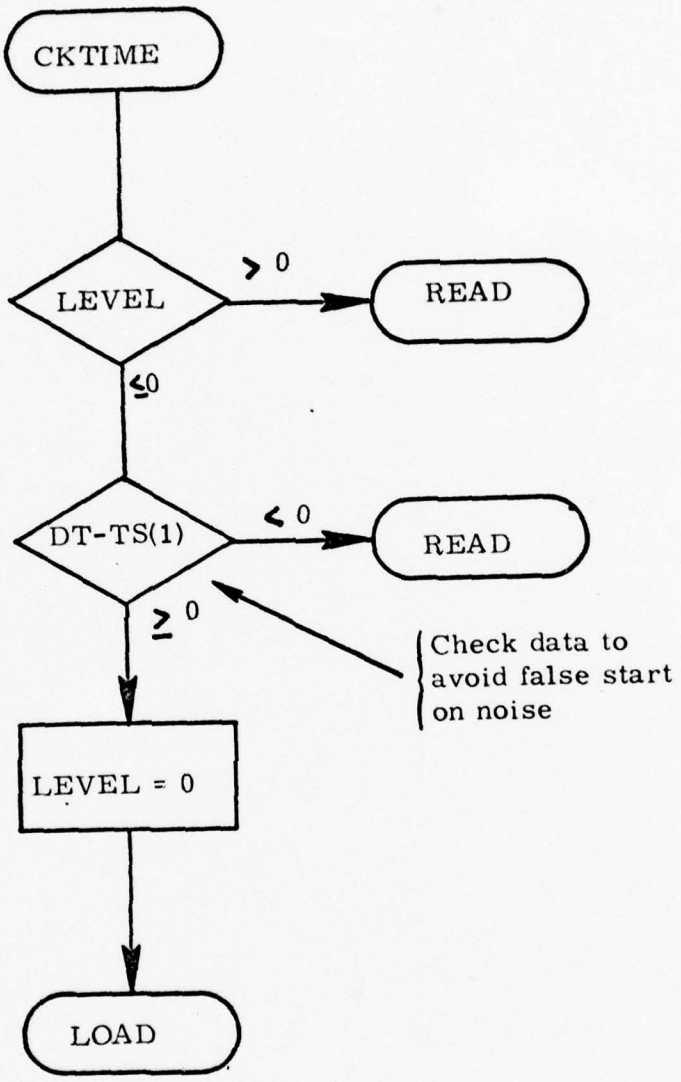

A
PAUSE 2
LOW
SENSE LINE 5
Pause for static check ok signal then give DDS a GO signal
HIGH
LAUNCH(I)=
Transfer initial conditions
BHOLD
IOUT = 16
Signal ICS sent
PRINT REAL-TIME STATUS
PAUSE 3
Enter real time with GO from DDS
RETURN
REALT
NPATH
=2
READ
=1
X

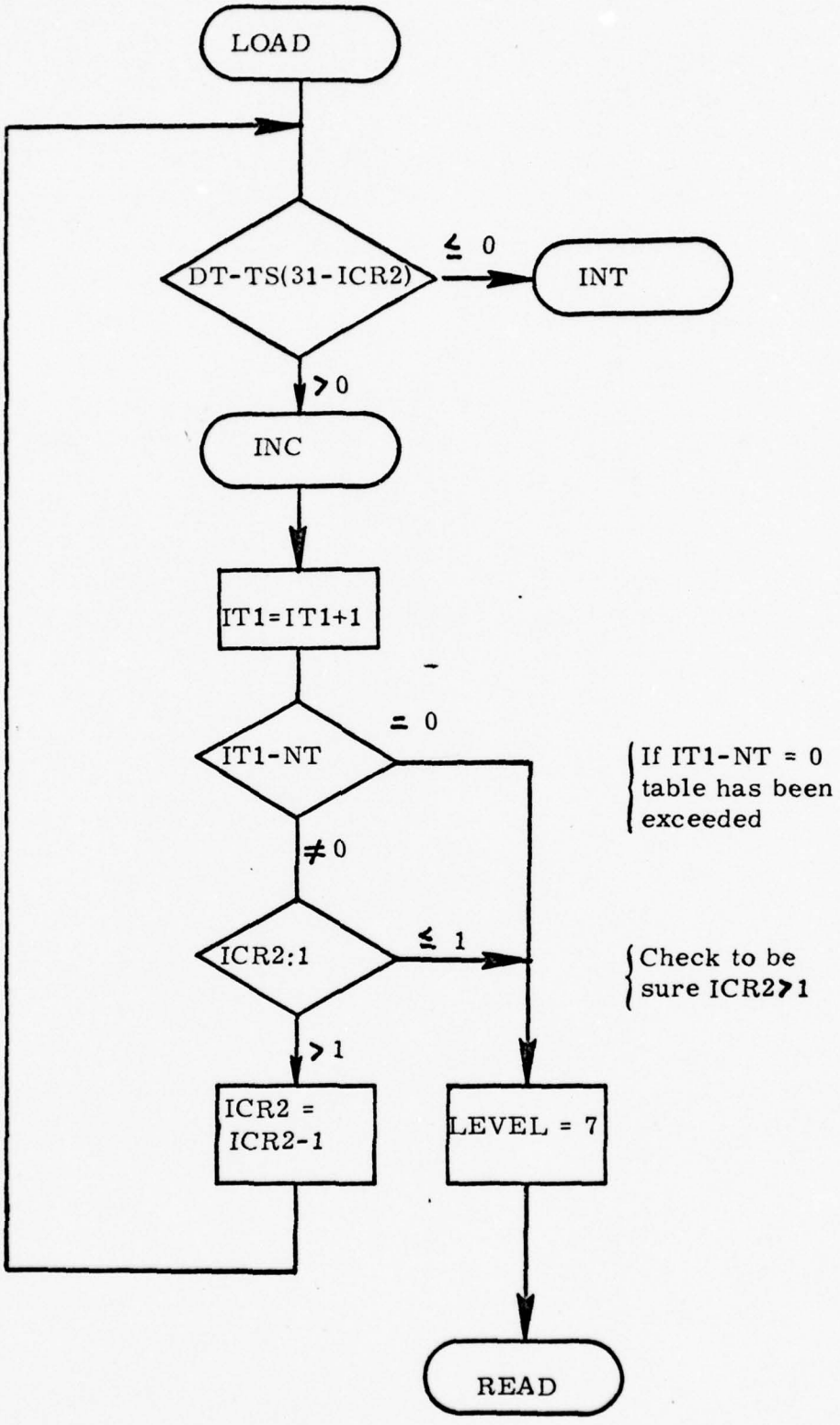

X
Wait for ramp up signal
SENSE LINE 6
LOW
HIGH
NPATH = 2
CALL SIMIDLE
READ
REALT
WMAN = 0
DX = ADIN(1)
DY = ADIN(2)
DZ = ADIN(3)
DT = ADIN(4)
XDOT = ADIN(5)
YDOT = ADIN(6)
ZDOT = ADIN(7)
XXX = ADIN(8)
YYY = ADIN(9)
ZZZ = ADIN(10)
Read and scale ADCs
B

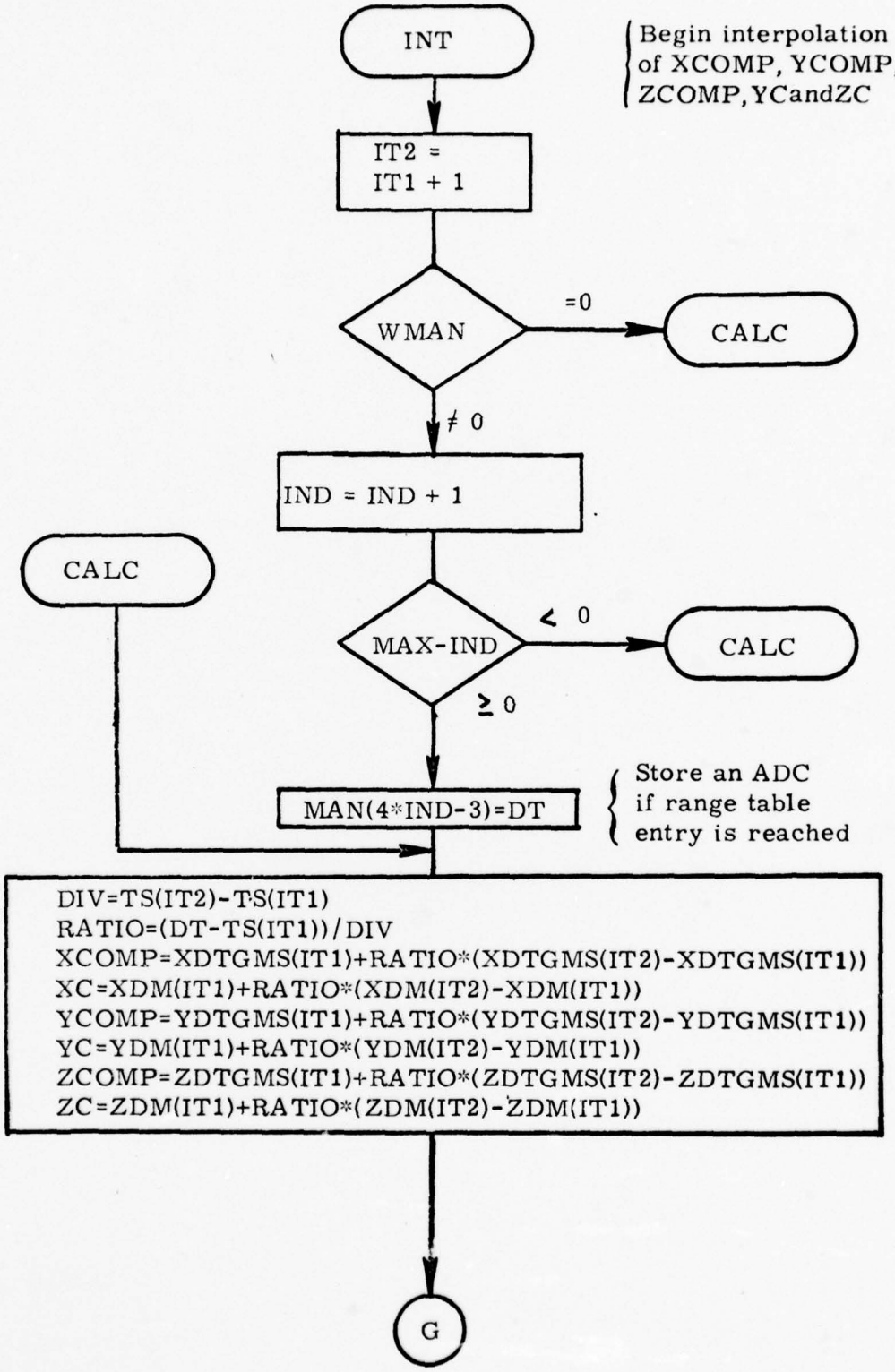

B
DXG = DX * SCALEP
DYG = DY * SCALEP
DZG = DZ * SCALEP
XDO = XDOT
YDO = YDOT
ZDO = ZDOT
Store DX, DY, DZ times SCALEP and XDOT, YDOT, ZDOT for diagnostic purposes
XXS(INDEX+1) -DX
> 0
TPRIME
≤ 0
Note DX and XXS are both negative no. prior to interrupt
Go to 3 channel mode if DX matches a range table entry
C

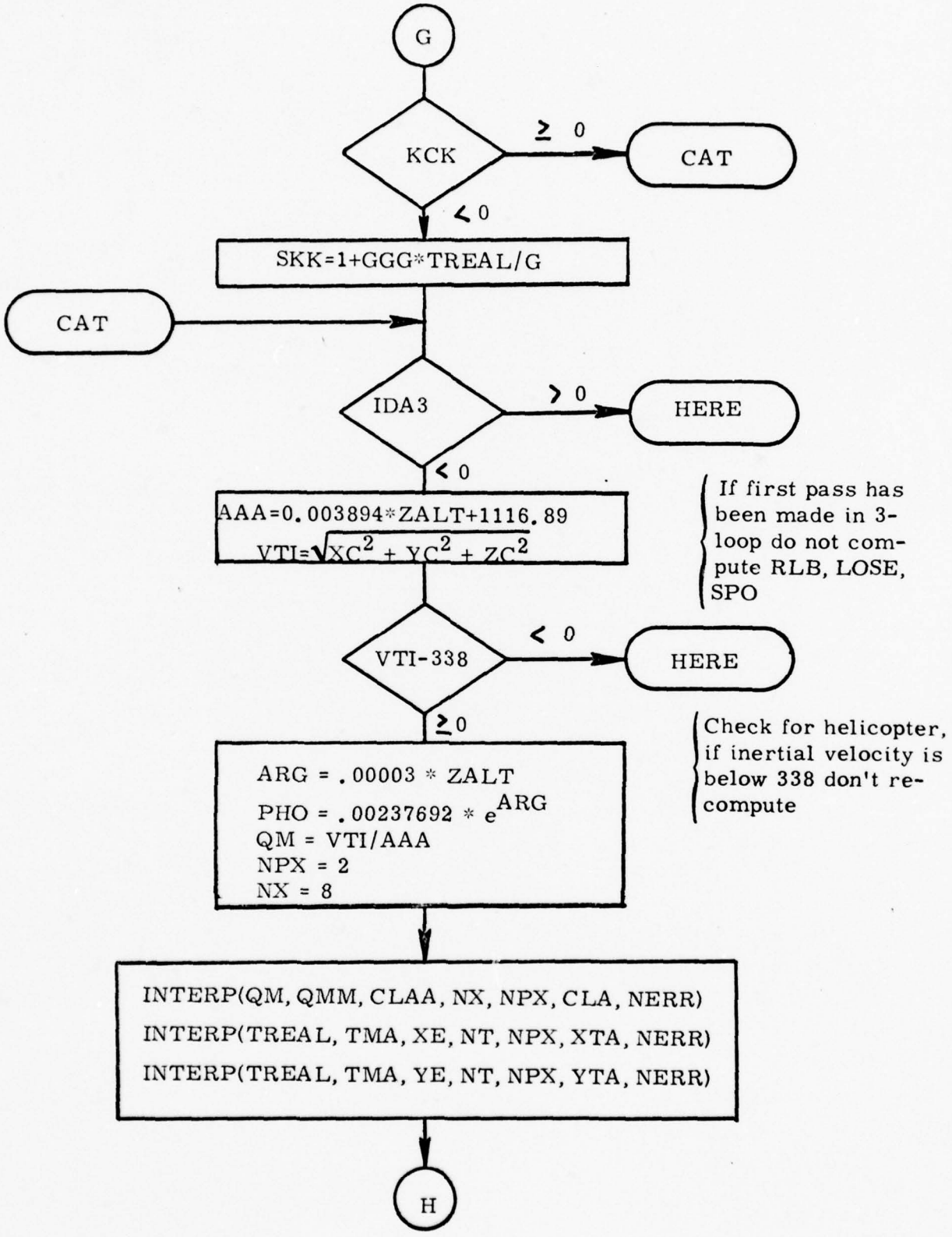

C
PPX (INDEX+1)=DX
PPY (INDEX+1)=DY
PPZ (INDEX+1)=DZ
TIME(INDEX+1)=DT
VMX(INDEX+1)=XDOT
VMY(INDEX+1)=YDOT
VMZ(INDEX+1)=ZDOT
Store ADCS for each range table entry reached
WMAN=7
IDA3=1
TPRIME
TREAL=DT* SCALET
Compute real time and variable scale factor
TREAL≤ G
FALSE
CKMISS
If t>t' go to ckmiss
TRUE
CKTIME

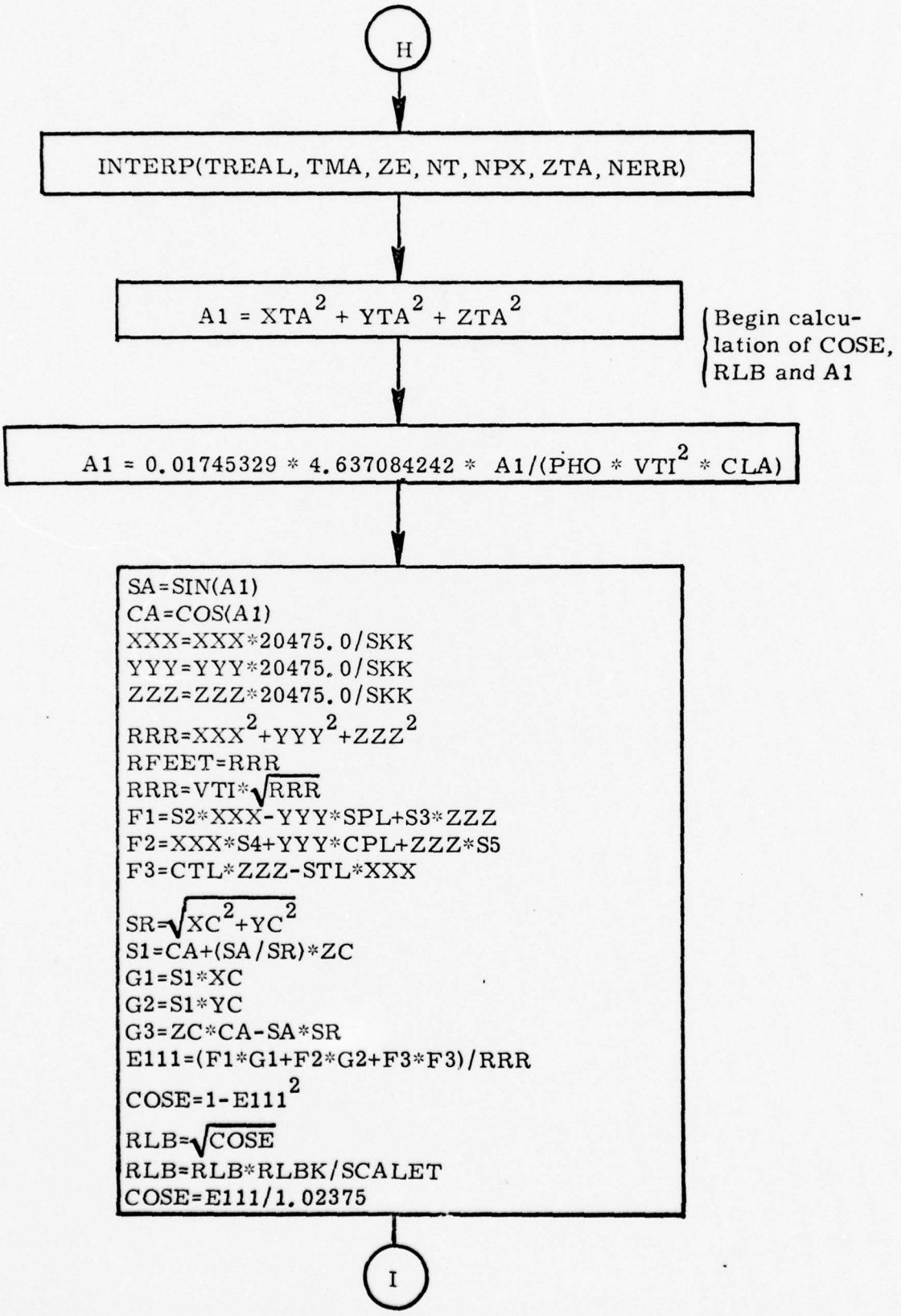

CKMISS
SKK = GGG +1
KCK = 1
MISS = XDOT * DX
MISS = MISS + DY *YDOT
MISS = MISS + DZ *ZDOT
Begin check for miss condition
MISS≤0
YES
CKTIME
NO
MISSED(1)=DX
MISSED(2)=DY
MISSED(3)=DZ
MISSED(4)=XDOT
MISSED(5)=YDOT
MISSED(6)=ZDOT
MISSED(7)=MISS
Store values that caused the miss
3000
SET SENSE LINE 7
HIGH
System hold except ramp down
STATUS = STAT(J)
CALL SIMSTOP
Check real-time status and return to batch processing
BHOLD

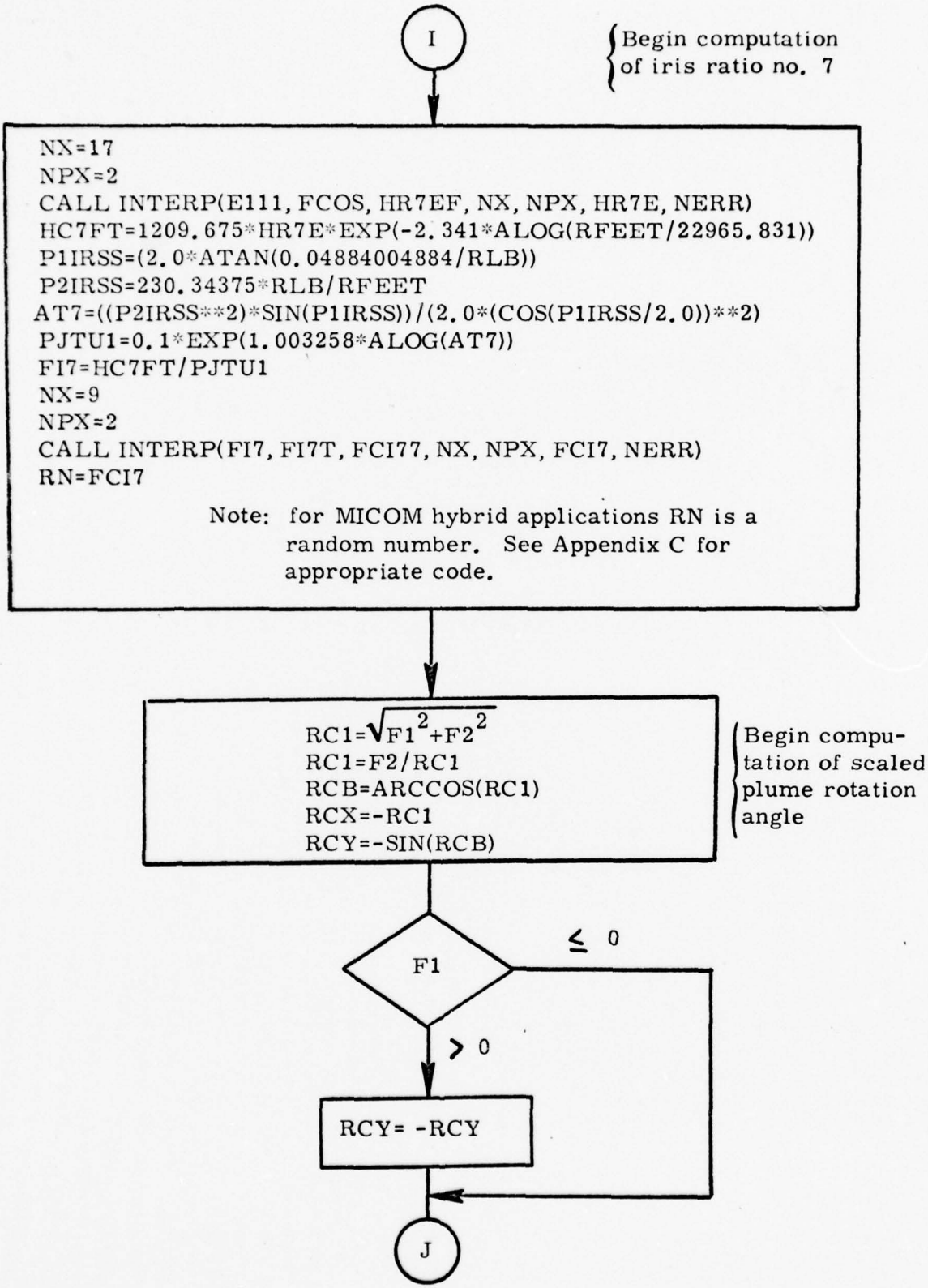

CKTIME
LEVEL
> 0
READ
≤0
DT-TS(1)
< 0
READ
≥ 0
Check data to avoid false start on noise
LEVEL = 0
LOAD

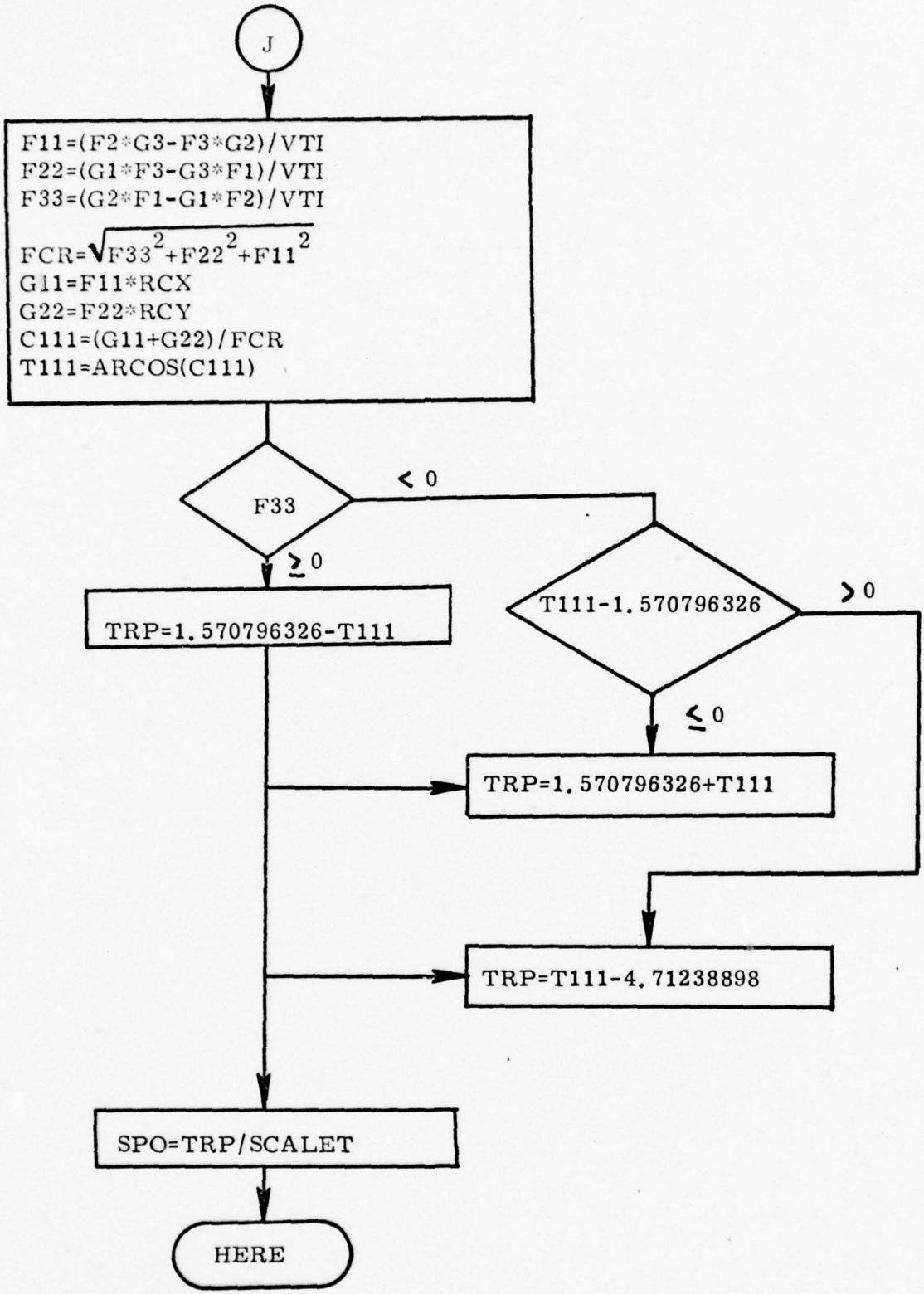

LOAD
DT-TS(31-ICR2)
≤ 0
INT
> 0
INC
IT1=IT1+1
IT1-NT
= 0
If IT1-NT = 0 table has been exceeded
≠ 0
ICR2:1
≤ 1
Check to be sure ICR2>1
> 1
ICR2 = ICR2-1
LEVEL = 7
READ

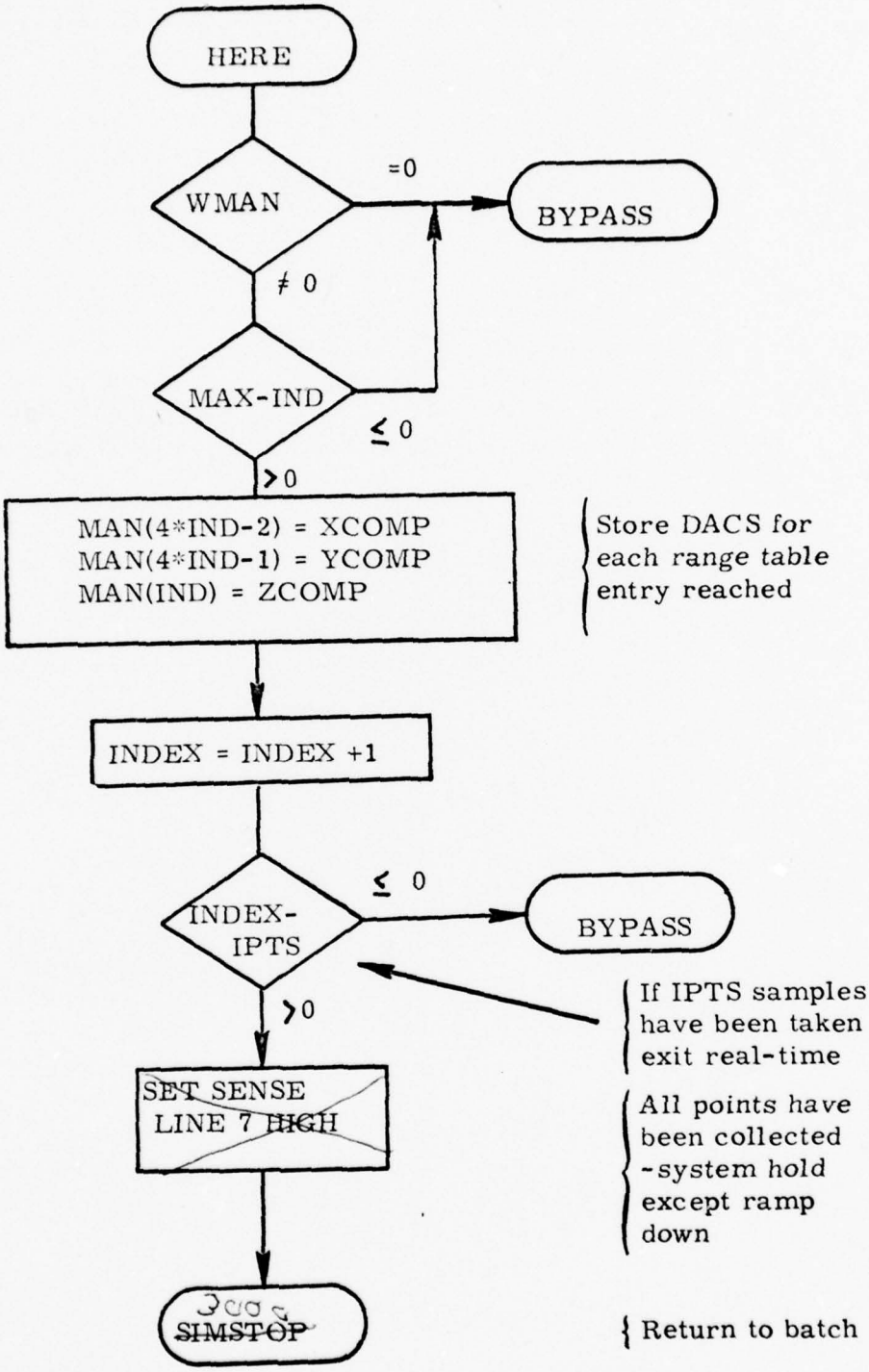

INT
Begin interpolation of XCOMP, YCOMP, ZCOMP, YCandZC
IT2 = IT1 + 1
WMAN
=0
CALC
≠ 0
IND = IND + 1
CALC
MAX-IND
< 0
CALC
≥ 0
MAN(4*IND-3)=DT
Store an ADC if range table entry is reached
DIV=TS(IT2)-TS(IT1)
RATIO=(DT-TS(IT1))/DIV
XCOMP=XDTGMS(IT1)+RATIO*(XDTGMS(IT2)-XDTGMS(IT1))
XC=XDM(IT1)+RATIO*(XDM(IT2)-XDM(IT1))
YCOMP=YDTGMS(IT1)+RATIO*(YDTGMS(IT2)-YDTGMS(IT1))
YC=YDM(IT1)+RATIO*(YDM(IT2)-YDM(IT1))
ZCOMP=ZDTGMS(IT1)+RATIO*(ZDTGMS(IT2)-ZDTGMS(IT1))
ZC=ZDM(IT1)+RATIO*(ZDM(IT2)-ZDM(IT1))
G


G
KCK
≥ 0
CAT
< 0
SKK=1+GGG*TREAL/G
CAT
IDA3
> 0
HERE
< 0
AAA=0.003894*ZALT+1116.89
VTI=√(XC² + YC² + ZC²)
If first pass has been made in 3-loop do not compute RLB, LOSE, SPO
VTI-338
< 0
HERE
≥0
Check for helicopter, if inertial velocity is below 338 don't re-compute
ARG = .00003 * ZALT
PHO = .00237692 * e^ARG
QM = VTI/AAA
NPX = 2
NX = 8
INTERP(QM, QMM, CLAA, NX, NPX, CLA, NERR)
INTERP(TREAL, TMA, XE, NT, NPX, XTA, NERR)
INTERP(TREAL, TMA, YE, NT, NPX, YTA, NERR)
H


H
INTERP(TREAL, TMA, ZE, NT, NPX, ZTA, NERR)
A1 = XTA^2 + YTA^2 + ZTA^2
Begin calculation of COSE, RLB and A1
A1 = 0.01745329 * 4.637084242 * A1/(PHO * VTI^2 * CLA)
SA=SIN(A1)
CA=COS(A1)
XXX=XXX*20475.0/SKK
YYY=YYY*20475.0/SKK
ZZZ=ZZZ*20475.0/SKK
RRR=XXX^2+YYY^2+ZZZ^2
RFEET=RRR
RRR=VTI*√RRR
F1=S2*XXX-YYY*SPL+S3*ZZZ
F2=XXX*S4+YYY*CPL+ZZZ*S5
F3=CTL*ZZZ-STL*XXX
SR=√XC^2+YC^2
S1=CA+(SA/SR)*ZC
G1=S1*XC
G2=S1*YC
G3=ZC*CA-SA*SR
E111=(F1*G1+F2*G2+F3*F3)/RRR
COSE=1-E111^2
RLB=√COSE
RLB=RLB*RLBK/SCALET
COSE=E111/1.02375
I


I
Begin computation of iris ratio no. 7
NX=17
NPX=2
CALL INTERP(E111, FCOS, HR7EF, NX, NPX, HR7E, NERR)
HC7FT=1209.675*HR7E*EXP(-2.341*ALOG(RFEET/22965.831))
P1IRSS=(2.0*ATAN(0.04884004884/RLB))
P2IRSS=230.34375*RLB/RFEET
AT7=((P2IRSS**2)*SIN(P1IRSS))/(2.0*(COS(P1IRSS/2.0))**2)
PJTU1=0.1*EXP(1.003258*ALOG(AT7))
FI7=HC7FT/PJTU1
NX=9
NPX=2
CALL INTERP(FI7, FI7T, FCI77, NX, NPX, FCI7, NERR)
RN=FCI7
Note: for MICOM hybrid applications RN is a random number. See Appendix C for appropriate code.
RC1=√F1^2+F2^2
RC1=F2/RC1
RCB=ARCCOS(RC1)
RCX=-RC1
RCY=-SIN(RCB)
Begin computation of scaled plume rotation angle
F1
≤ 0
> 0
RCY= -RCY
J


J
F11=(F2*G3-F3*G2)/VTI
F22=(G1*F3-G3*F1)/VTI
F33=(G2*F1-G1*F2)/VTI
FCR=√(F33^2+F22^2+F11^2)
G11=F11*RCX
G22=F22*RCY
C111=(G11+G22)/FCR
T111=ARCOS(C111)
F33
< 0
≥ 0
T111-1.570796326
> 0
≤ 0
TRP=1.570796326-T111
TRP=1.570796326+T111
TRP=T111-4.71238898
SPO=TRP/SCALET
HERE


HERE
WMAN
=0
BYPASS
≠ 0
MAX-IND
≤ 0
>0
MAN(4*IND-2) = XCOMP
MAN(4*IND-1) = YCOMP
MAN(IND) = ZCOMP
Store DACS for each range table entry reached
INDEX = INDEX +1
INDEX-IPTS
≤ 0
BYPASS
>0
If IPTS samples have been taken exit real-time
SET SENSE LINE 7 HIGH
All points have been collected -system hold except ramp down
SIMSTOP
Return to batch

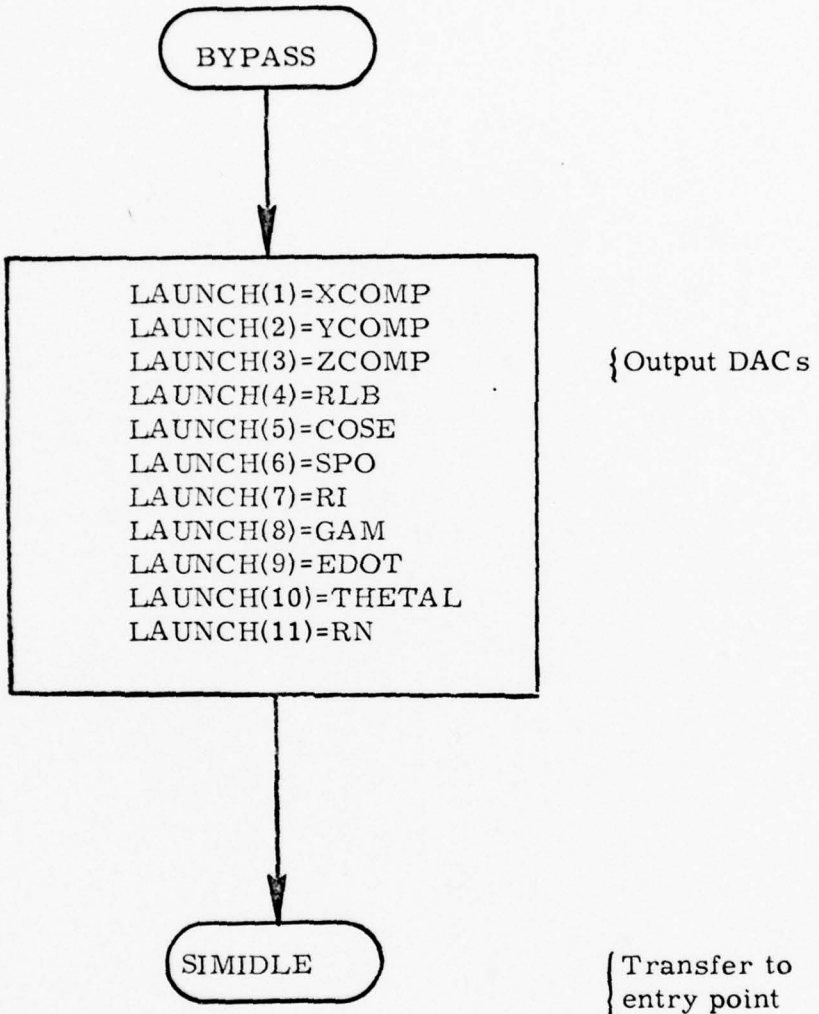

BYPASS
LAUNCH(1)=XCOMP
LAUNCH(2)=YCOMP
LAUNCH(3)=ZCOMP
LAUNCH(4)=RLB
LAUNCH(5)=COSE
LAUNCH(6)=SPO
LAUNCH(7)=RI
LAUNCH(8)=GAM
LAUNCH(9)=EDOT
LAUNCH(10)=THETAL
LAUNCH(11)=RN
{Output DACs
SIMIDLE
{Transfer to entry point READ

TABLE 2.1

DESCRIPTION OF PROGRAM SYMBOLS

VARIABLE	PROGRAM SYMBOL*	DESCRIPTION
———	SPO	Scaled plume rotation angle
T_{RP}	TRP	Plume rotation angle (RAD.)
θ_L	THETAL	Initial elevation angle of target (DEG.)
t	TREAL	Real time (SEC.)
g	———	Gravitational constant
x_E	XE	Target aerodynamic acceleration table in earth fixed coordinates
y_E	YE	
z_E	ZE	
α	A1	Target angle of attack (RAD.)
t'	G	Time at approximately 1000 feet to go
C_{LA}	CLA	Aerodynamic lift coefficient due to angle of attack
$\overline{V}_{TI}$	———	Target inertial velocity
ℓ/B	RLB	Apparent plume length to breadth ratio
E	———	Subscript which denotes earth fixed coordinates

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
GN	———	Subscript which denotes generalized target coordinates
F	———	Subscript which denotes target fixed coordinates
G	———	Subscript which denotes guidance coordinates
L	———	Subscript which denotes launch coordinates
R_i, R	RI, RFEET	Initial range (FT.)
———	EDOT	Scaled γ/t'
———	GAM	
ρ	PHO	Air density (Slugs/FT.^{3})
γ_{min}	GGG	Minima of the overload function
Scalet	SCALET	Scale Factor = 10.2375
K	SKK	Variable scale factor, $K = 1 + \frac{\gamma t}{t'}$, $t \leq t'$; $1+\gamma$, $t > t'$
———	XX	Range table for data collection
———	XXS	Scaled range table (XX)

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
I_{r7}	RN	For MICOM Hybrid, a uniform random number, RN ϵ (-1, 1) For IRSS, iris ratio number 7
Cos (ϵ)	COSE	Cosine of angle between LOS and center line of target
$x_{ML}-x_{TL}$	DX	Scaled xxx, yyy, zzz
$y_{ML}-y_{TL}$	DY	
$z_{ML}-z_{TL}$	DZ	
t	DT	Real time
$x_{ML}-x_{TL}$	XXX	x-Missile minus x-target position (FT.)
$y_{ML}-y_{TL}$	YYY	y-Missile minus y-target position (FT.)
$z_{ML}-z_{TL}$	ZZZ	z-Missile minus z-target position (FT.)
$\dot{x}_{ML}-\dot{x}_{TL}$	XPOT	x-Missile minus x-target velocity (FT./SEC.)
$\dot{y}_{ML}-\dot{y}_{TL}$	YDOT	y-Missile minus y-target velocity (FT./SEC.)
$\dot{z}_{ML}-\dot{z}_{TL}$	ZDOT	z-Missile minus z-target velocity (FT./SEC.)

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
$\dot{x}_{TL}$	XDTGMS	Target manuvers and manuver time
$\dot{y}_{TL}$	YDTGMS	
$\dot{z}_{TL}$	ZDTGMS	
ℓ	———	Apparent plume length (FT.)
B	———	Apparent plume breadth (FT.)
β	———	$\cos^{-1}(L_y / \sqrt{L_x^2 + L_y^2})$
h	ZALT	Altitude above sea level
———	XCOMP	Interpolated value of XDTGMS
———	YCOMP	Interpolated value of YDTGMS
———	ZCOMP	Interpolated value of ZDTGMS
$\dot{x}_{TE}$	XC	Interpolated value of XDM
$\dot{y}_{TE}$	YC	Interpolated value of YDM
$\dot{z}_{TE}$	ZC	Interpolated value of ZDM

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
———	XDM	Target velocity table selected is stored in XDM, YDM, ZDM respectively
———	YDM	
———	ZDM	
———	TIME	Storage for missile position, velocity and time in the region of pre-specified range table entries
———	PPX	
———	PPY	
———	PPZ	
———	VMX	
———	VMY	
———	VMZ	
———	NPTS	Number of points in range table
———	MISSED	An array of values representing miss conditions
	T111	Plume rotation angle
	———	Time = 0 on target trajectory

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
x_G	———	x-Missile minus x-target (in launch coordinates)
y_G	———	y-Missile minus y-target (in launch coordinates)
z_G	———	z-Missile minus z-target (in launch coordinates)
VT	VTI	Target inertial velocity
———	XTA	Interpolated from XE at TREAL
———	YTA	Interpolated from YE at TREAL
———	ZTA	Interpolated from ZE at TREAL
$\overline{C}_L$	———	Longitudinal center line of target
C_{Lx}	———	Components of $\overline{C}_L$
C_{Ly}	———	
C_{Lz}	———	
$\overline{L}_{os}$	———	Line of sight vector

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
L_x	———	Components of $\bar{L}_{os}$
L_y	———	
L_z	———	
x_{ML}	———	x-Missile position in launch coordinates
y_{ML}	———	y-Missile position in launch coordinates
z_{ML}	———	z-Missile position in launch coordinates
x_{TL}	———	x-target position in launch coordinates
y_{TL}	———	y-Target position in launch coordinates
z_{TL}	———	z-Target position in launch coordinates
x_{TE}	———	x-Position of target in inertial coordinates
y_{TE}	———	y-Position of target in inertial coordinates
z_{TE}	———	z-Position of target in inertial coordinates

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
———	XDTGO	Initial values of XDTGMS, YDTGMS, ZDTGMS
———	YDTGO	
———	ZDTGO	
H_{c7}	———	Commanded irradiance for plume
f	19.5	Focal length of projection lens (CM)
$H\Big\vert_{\epsilon}^{R=7000}$	HR7E	Irradiance at 7000 meters (W/CM^2)
P_1	P1IRSS	Vertex angle of plume
P_2	P2IRSS	Apparent altitude of plume (IN.)
A_{t7}	AT7	Area of plume transparency ($IN.^2$)
$J_{tu1,\,plume}$	PJTU1	Radiant intensity used as function of plume transparency area taken from calibration in IRSS
$T_{f,\,plume}$	0.3	Neutral density transmission factor for plume

VARIABLE	PROGRAM SYMBOL	DESCRIPTION
———	DXG	Dummy storage location for current value of DX, DY, DZ, XDOT, YDOT AND ZDOT respectively. These variables are returned to the main program for diagnostic purposes
———	DYG	
———	DZG	
———	XDO	
———	YDO	
———	ZDO	
———	LAUNCH	Array of DAC variables
———	ADIN	Array of ADC variables
———	IWRITE	Flag for writing comments if IWRITE = 1
———	LEVEL	Status of maneuver -7 implies not in real time 0 implies in real time +7 implies target traj table exceeded
———	WMAN	Flag, =1 implies collect data, =7 implies don't collect data
———	MAN	Array for storing target manuever table
———	MAX	Maximum points collected

VARIABLES	PROGRAM SYMBOL	DESCRIPTION
———	F1, F2, F3	Intermediate variables in the computation of COS (ϵ)
———	G1, G2, G3	
———	RRR	Range-to-go
———	IND	Index of XCOMP, YCOMP and ZCOMP arrays
———	INDEX	Index of range table entries
———	KCK	Flag, =1 implies $t > t'$, = -1 implies $t \leq t'$
———	ICR2	Counter, initial value = length of array TS minus one
———	NPTS	Number of points in range table
———	IDA3	Logical variable, = -1 implies not yet in 3-loop, =1 implies first pass in 3-loop mode
———	XMISS	Array to save ADCS under miss condition (See MISSED)

* In some cases, program symbols are used for intermediate storage or may have scale factors other than those implied. For further information see Appendix A.

2.2 REAL TIME I/O CONSIDERATIONS

Phase 1 conversion of the STINGER real time program requires real time analog and discrete communication between the AD/4 analog computer and the CDC/6600 digital computer. This high speed communications link is provided by the DADIOS. Specific requirements include;

- 10-Analog to digital channels
- 11-Digital to analog channels
- 1-Discrete control line (2 bits required)
- 1-Discrete sense line (2 bits required)

In Figure 2.3 this arrangement of I/O is depicted graphically.

The actual assignment of these analog and digital signals is described in Tables 2.2 - 2.5. These tables provide a cross reference of trunkline assignment, variable name and scale factors.

Replacement of the IBM/DOS interface with new DADIOS equipment required scale factor modifications to both the ADC and DAC channels. This modification was implemented in the digital computer code as a constant scale factor of 1/1.02375 multiplied times each ADC channel and a constant scale factor of .102375 multiplied times each DAC channel. The rationale for this scale factor change is presented in Figures 2.4 and 2.5.

In Figure 2.6 the DADIOS patching required for the STINGER simulation is presented. This figure ndicates that a total of 32 analog lines are available although fewer lines are actually used. Figure 2.6 similarly depicts a total of 32 discrete line available, where in fact only four lines are currently utilized.

Preliminary checkout of the DADIOS/CDC-6600/DDS/AD-4 system can be performed in an open loop test with the STINGER real time skeleton test program. This program provides a check on the operational status of all equipment involved in the phase 1 conversion task. A description of this program is given in Appendix D.

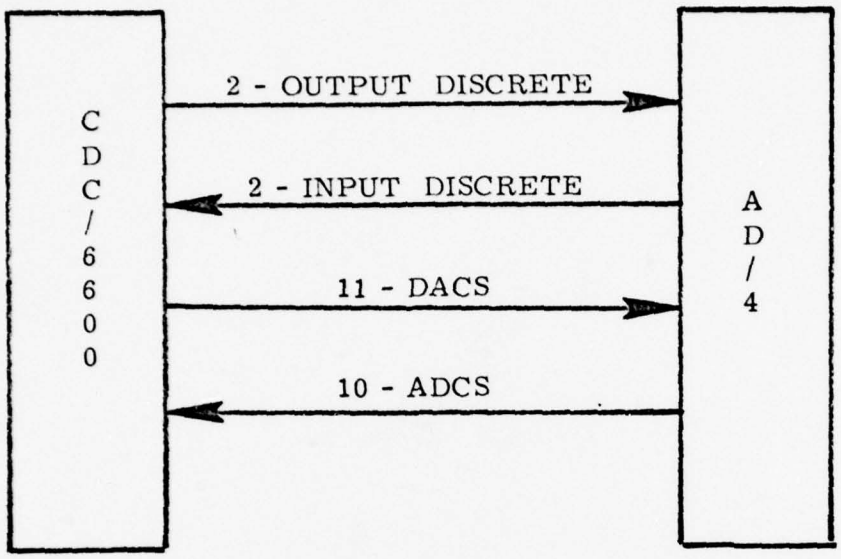
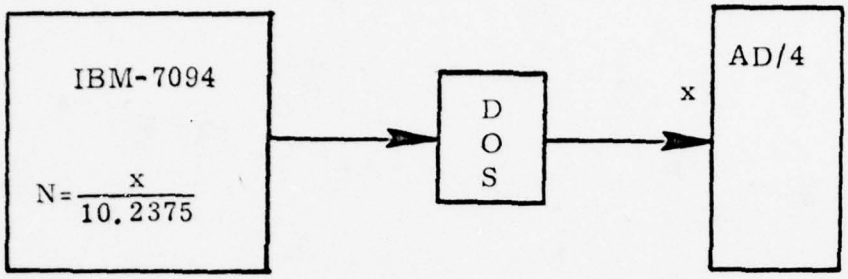


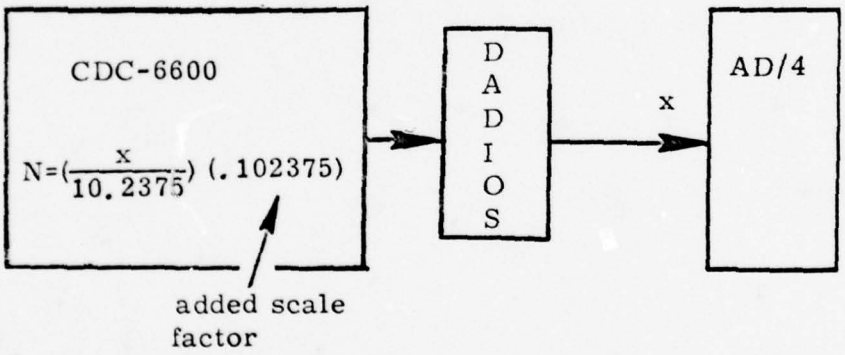
FIGURE 2.3
PHASE 1 REAL TIME I/O REQUIREMENTS

The DOS provided the following scaling



where $-10 \leq x \leq +10$

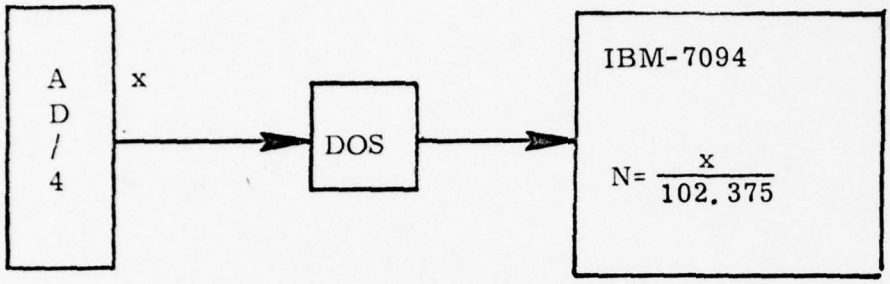
Now, the CDC-6600 must provide the following scaling



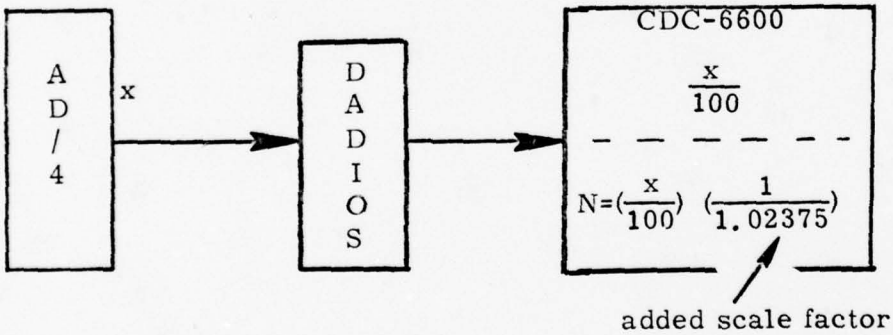
Therefore, digital output information should be premultiplied by .102375 before being presented to the DACs.

FIGURE 2.4 DAC SCALE FACTOR ADJUSTMENT

The DOS provided the following scaling



Now, the CDC-6600 must provide the following scaling



Therefore, each ADC should be premultiplied by (1/1.02375) before being used by the STINGER real time digital program

Figure 6. ADC scale factor adjustments

FIGURE 2.5 ADC SCALE FACTOR ADJUSTMENT

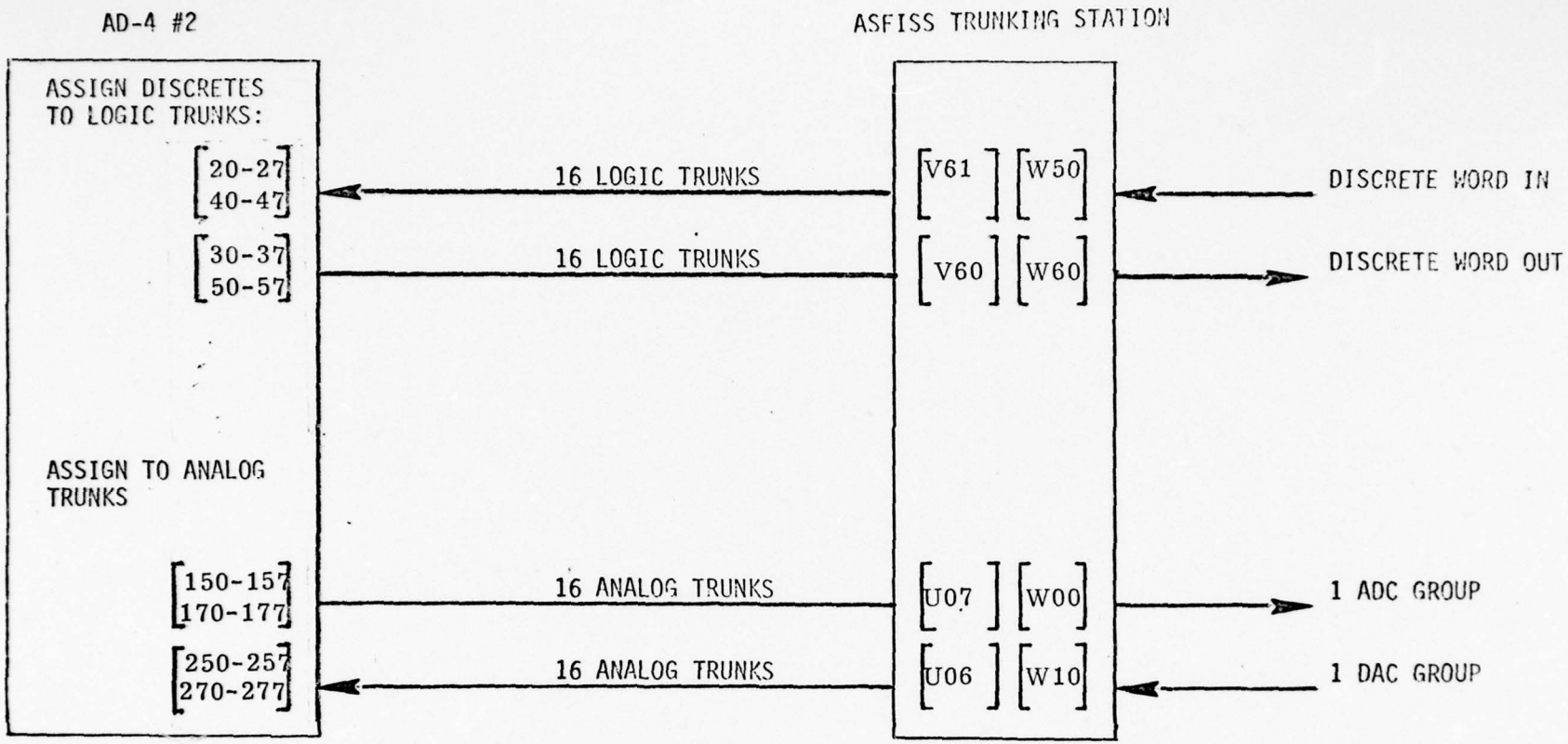


FIGURE 2.4

PHASE 1 TRUNKLINE REQUIREMENTS FOR STINGER SIMULATION

TABLE 2.2

PHASE 1 CONTROL LINE ASSIGNMENT
(CDC OUTPUT DISCRETES)

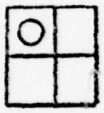
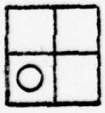
CDC-6600 VARIABLE	CDC-6600 BIT	AD/4 # 2 TRUNK LINE	PURPOSE/ACTION
CTRL0-CTRL3	0 - 3	TR57-TR54	Not in use
CTRL4 	4	TR53	Is set high after ICs have been sent to analog. It is also the signal that starts the analog (SYS-OP) for MICOM hybrid
CTRL5, CTRL6	5, 6	TR52, TR51	Not in use
CTRL7 	7	TR50	Is set high when the simulation is terminated and the analog should go into system hold. Termination can occur for "missed target" or "all data collected"
CTRL8-CTRL15	8 - 15	TR37-TR30	Not in use

TABLE 2.3

PHASE 1 SENSE LINE ASSIGNMENT
(CDC INPUT DISCRETES)

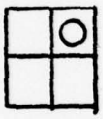
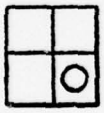
CDC-6600 VARIABLE	CDC-6600 BIT	AD/4 # 2 TRUNK LINE	PURPOSE/ACTION
SL0-SL3	0 - 3	TR47-TR44	Not in use
SL4	4	TR43	Not in use
SL5 	5	TR42	Set high by the static check OK switch on AD/4. This but must be high to send ICs to analog
SL6 	6	TR41	Set high by the static check OK switch on the AD/4 for MICOM Hybrid, or set high by the ramp up circuit for IRSS hybrid when initial IRSS conditions are attained
SL7-SL15	7 - 15	TR40 and TR27-TR20	Not in use

TABLE 2.4

PHASE 1 ADC ASSIGNMENT

ANALOG VARIABLE NAME	AD/4 ANLG. VOLTAGE $-100 \leq V \leq +100$	DIGITAL FRACTION $-1 \leq N < +1$	CDC/6600 ADC ASSIGNMENT	AD/4 TRUNK LINE ASSIGNMENT
DX	$\frac{K(X_{ML}-X_{TL})}{20}$	$\frac{K(X_{ML}-X_{TL})}{(20)(102.375)}$	2	151
DY	$\frac{K(Y_{ML}-Y_{TL})}{20}$	$\frac{K(Y_{ML}-Y_{TL})}{(20)(102.375)}$	3	152
DZ	$\frac{K(Z_{ML}-Z_{TL})}{20}$	$\frac{K(Z_{ML}-Z_{TL})}{(20)(102.375)}$	4	153
DT	10t	$\frac{10t}{102.375}$	5	154
XDOT	$\frac{\dot{X}_{ML}-\dot{X}_{TL}}{200}$	$\frac{\dot{X}_{ML}-\dot{X}_{TL}}{(200)(102.375}$	6	155
YDOT	$\frac{\dot{Y}_{ML}-\dot{Y}_{YL}}{200}$	$\frac{\dot{Y}_{ML}-\dot{Y}_{TL}}{(200)(102.375)}$	7	156
ZDOT	$\frac{\dot{Z}_{ML}-\dot{Z}_{TL}}{200}$	$\frac{\dot{Z}_{ML}-\dot{Z}_{TL}}{(200)(102.375)}$	8	157

ANALOG VARIABLE NAME	AD/4 ANLG. VOLTAGE $-100 < V < +100$	DIGITAL FRACTION $-1 < N < +1$	CDC/6600 ADC ASSIGNMENT	AD/4 TRUNK LINE ASSIGNMENT
XXX	$\frac{K(X_{ML}-X_{TL})}{200}$	$\frac{K(X_{ML}-X_{TL})}{(200)(102.375)}$	9	170
YYY	$\frac{K(Y_{ML}-Y_{TL})}{200}$	$\frac{K(Y_{ML}-Y_{TL})}{(200)(102.375)}$	10	171
ZZZ	$\frac{K(Z_{ML}-Z_{TL})}{200}$	$\frac{K(Z_{ML}-Z_{TL})}{(200)(102.375)}$	12	173

TABLE 2.5

PHASE 1 DAC ASSIGNMENT

DIGITAL VARIABLE NAME	DIGITAL FRACTION $-1<N<+1$	AD/4 ANLG. VOLTAGE $-10<V<+10$	CDC/6600 DAC ASIGN.	AD/4 TRUNK LINE ASSIGNMENT
XDTGO, XCOMP	$\frac{\dot{X}_{TL}}{(200)(10.375)}$	$\frac{\dot{X}_{TL}}{200}$	1	250
YDTGO, YCOMP	$\frac{\dot{Y}_{TL}}{(200)(10.375}$	$\frac{\dot{Y}_{TL}}{200}$	2	251
ZDTGO, ZCOMP	$\frac{\dot{Z}_{TL}}{(200)(10.375)}$	$\frac{\dot{Z}_{TL}}{200}$	3	252
RLB	$\frac{(\ell/B)}{10.2375}$	ℓ/B	4	253
COSE	$\frac{10\cos(\epsilon)}{10.375}$	$10\cos(\epsilon)$	5	254
SPO	$\frac{T_{RP}}{10.2375}$	T_{RP}	6	255
RI	$\frac{R_i}{(2000)(10.375)}$	$\frac{R_i}{2000}$	7	256

DIGITAL VARIABLE NAME	DIGITAL FRACTION $-1 < N < +1$	AD/4 ANLG. VOLTAGE $-10 < V < +10$	CDC/6600 DAC ASIGN.	AD/4 TRUNK LINE ASSIGNMENT
GAM	$\frac{t'}{10.2375}$	t'	8	257
EDOT	$\frac{(\gamma/t')}{(.6)(10.2375)}$	$\frac{5(\gamma/t')}{3}$	9	270
THETAL	$\frac{\theta_L}{(10)(10.2375)}$	$\frac{\theta_L}{10}$	10	271
RN	$\frac{i_7}{10.2375}$	i_7	12	273
RN*	$\frac{R_n}{10.2375}$	R_n	12	273

* FOR MICOM HYBRID APPLICATIONS

3.0 ADC CHECKOUT PROGRAM FOR AD-4 #2

3.1 PURPOSE

The program reads 16 ADC channels (220-363) on AD-4 number 2 and compares the values read to either of two references: 1. stored references entered with the program or 2. references obtained by a single scan of the ADC channels. Selection of this option is under sense switch control. When a value read back is outside a tolerance specified by the user (default value for tolerance is 1 bit) the program will record an error, note its magitude, and record the actual number read in a high or low failure buffer. The program prints out the accumulated error data every 30 seconds. A sample printout preceeds the program listing.

3.2 USAGE

A. To load program into memory do the following:

1. Put Binary program deck behind universal loader.
2. Ready card reader and printer.
3. On the console, push Idle then Reset.
4. Clear register lights.
5. Hold down Clear and Clear Flags for 1 second.
6. Set Sense Switch 4.
7. Press Reset, Run, and Cards.

B. To execute the program which has been previously loaded do the following:

1. Press Idle, Reset.
2. With register display set to B, enter 00160200.
3. Clear AD-4 Interface.
4. Press Run.

C. To select reference mode, set SS2. This will cause the program to compare values read to those stored in the buffer beginning at location 65240. Default values for these references are zero.

D. To halt, set SS6. To continue, turn SS6 off and hit Run.

E. To suppress print out, set SS1.

F. To modify tolerance, load value desired in location 65561.

G. To modify print period, load value in RCOUNT (e.g. $20,000_8$ prints out every half hour) at location 65560.

3.3 HARDWARE USED

- SDS 9300 computer, printer and card reader
- AD-4/9300 Interface
- AD-4 number 2

FIGURE 3.1 ADC CHECK PROGRAM PRINTOUT

00060040	00000000
00060041	00000000
00060042	00000000
00060043	00000000
00060044	00000000
00060045	00000000
00060046	00000000
00060047	00000000
00060050	00000000
00060051	00000000
00060052	00000000
00060053	00000000
00060054	00000000
00060055	00000000
00060056	00000000
00060057	00000000
00060060	00000000
00060061	00000000
00060062	00000000
00060063	00000000
00060064	00000000
00060065	00000000
00060066	00000000
00060067	00000000
00060070	00000000
00060071	00000003

00060040	00000206
00060041	00000552
00060042	00000112
00060043	00000000
00060044	00007662
00060045	00022544
00060046	00000551
00060047	00000106
00060050	00000000
00060051	00000000
00060052	00000000
00060053	00000000
00060054	00000000
00060055	00000017
00060056	00000000
00060057	00000000
00060060	00000000
00060061	00000000
00060062	00000000
00060063	00000000
00060064	00000000
00060065	00000000
00060066	00000000
00060067	00000000
00060070	00000000
00060071	00000004

```
00060040    00000000
00060041    00000000
00060042    00000000
00060043    00000000
00060044    00000000
00060045    00000000
00060046    00000000
00060047    00000000
00060050    00000000
00060051    00000000
00060052    00000000
00060053    00000000
00060054    00177767
00060055    00000000
00060056    00000000
00060057    00000000
00060060    00000000
00060061    00000000
00060062    00000000
00060063    00000000
00060064    00000000
00060065    00000000
00060066    00000000
00060067    00000000
00060070    00000000
00060071    00000005
```

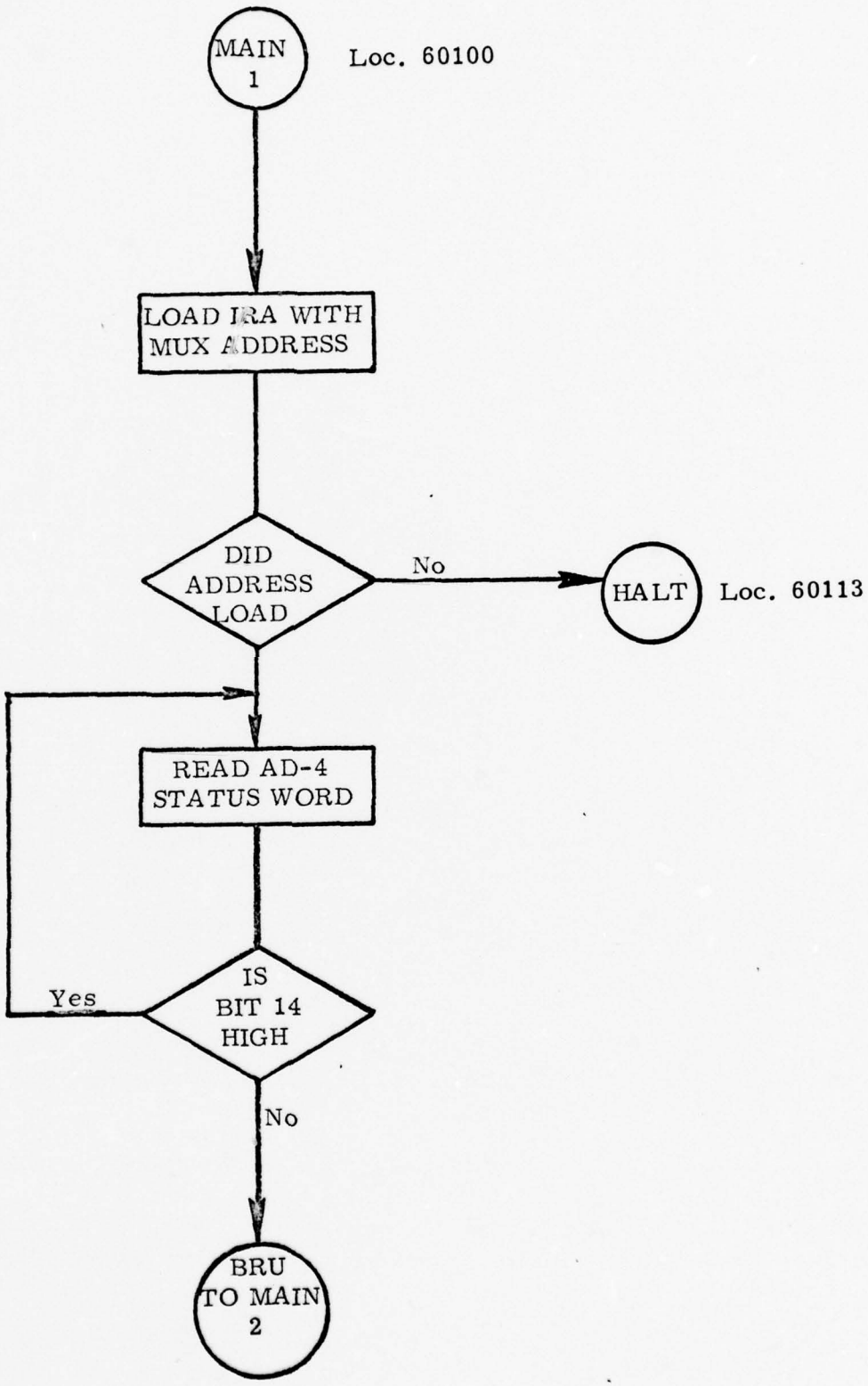
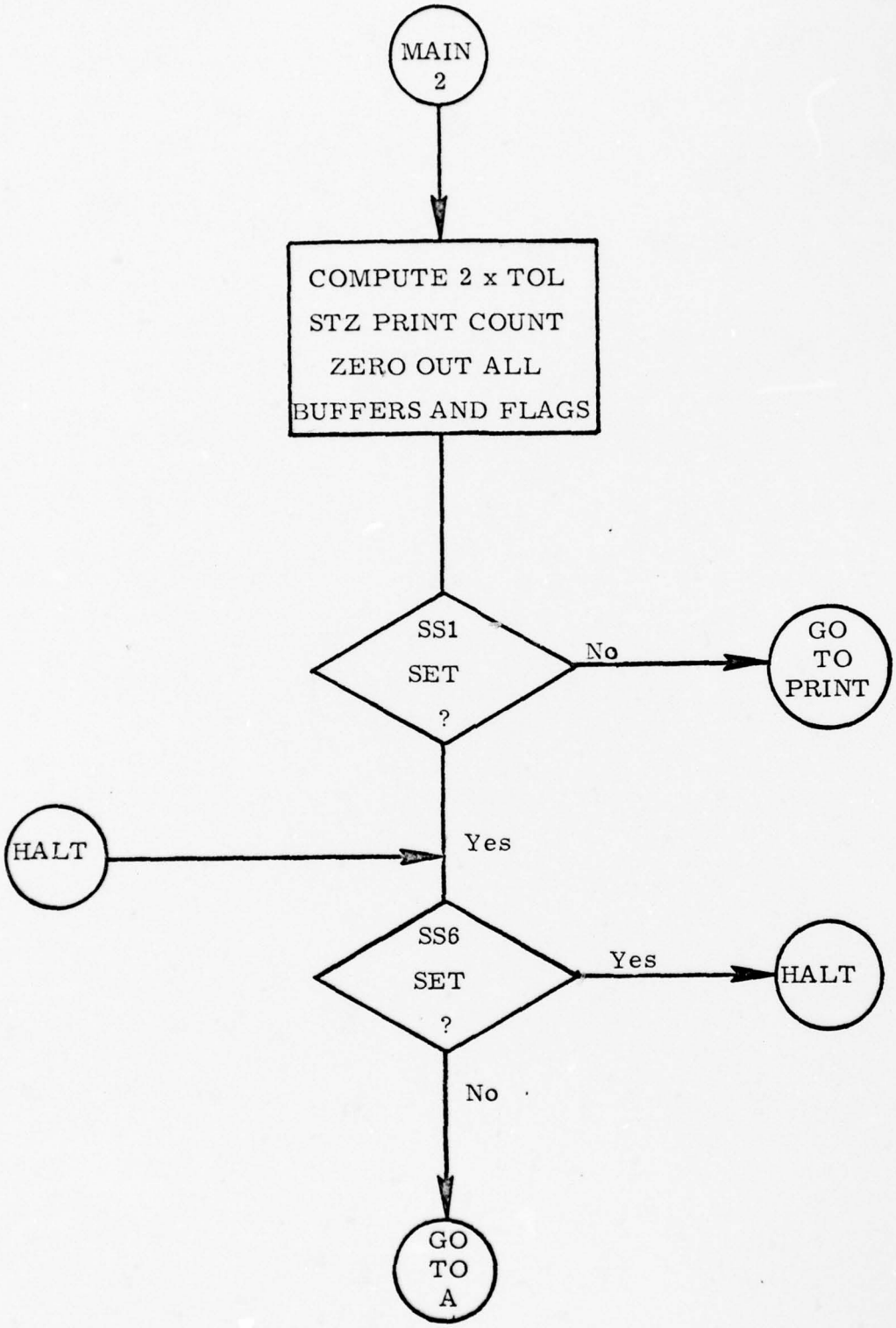
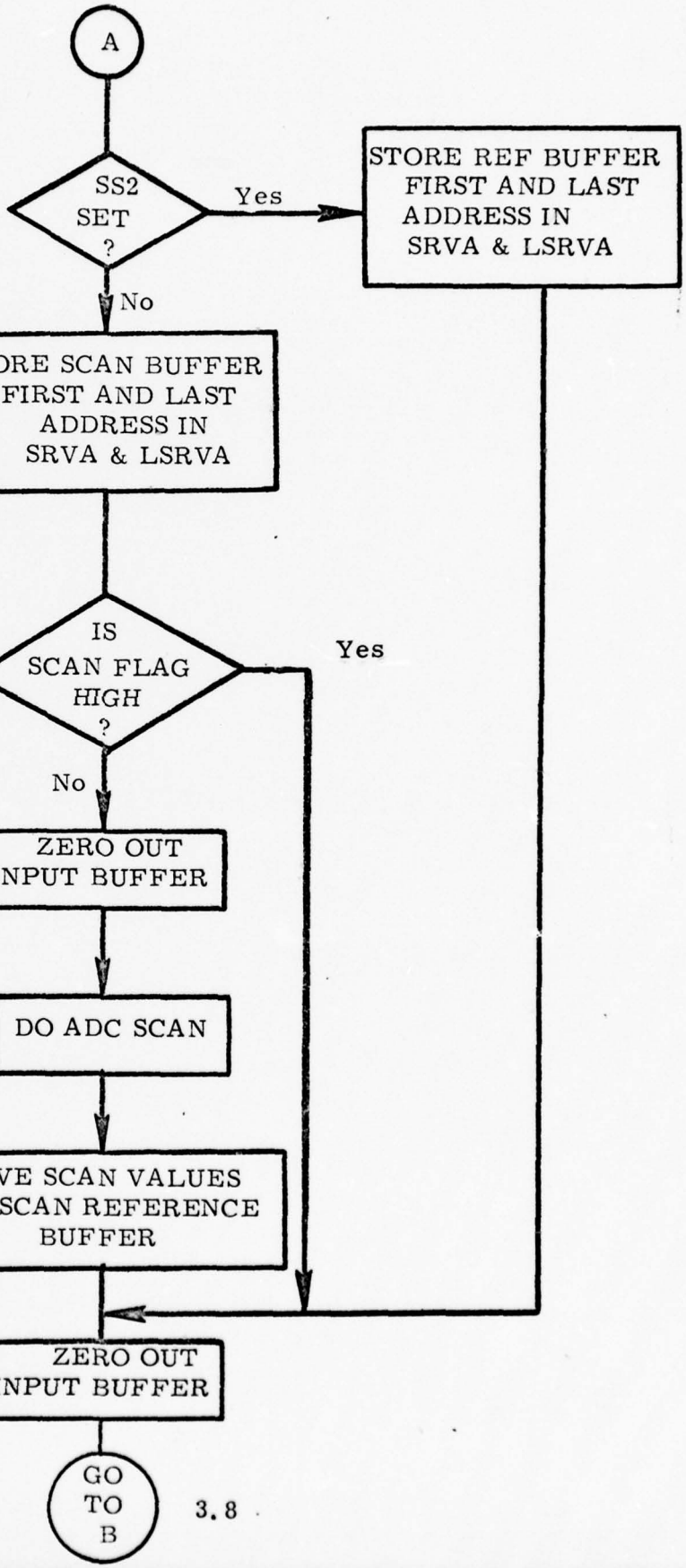
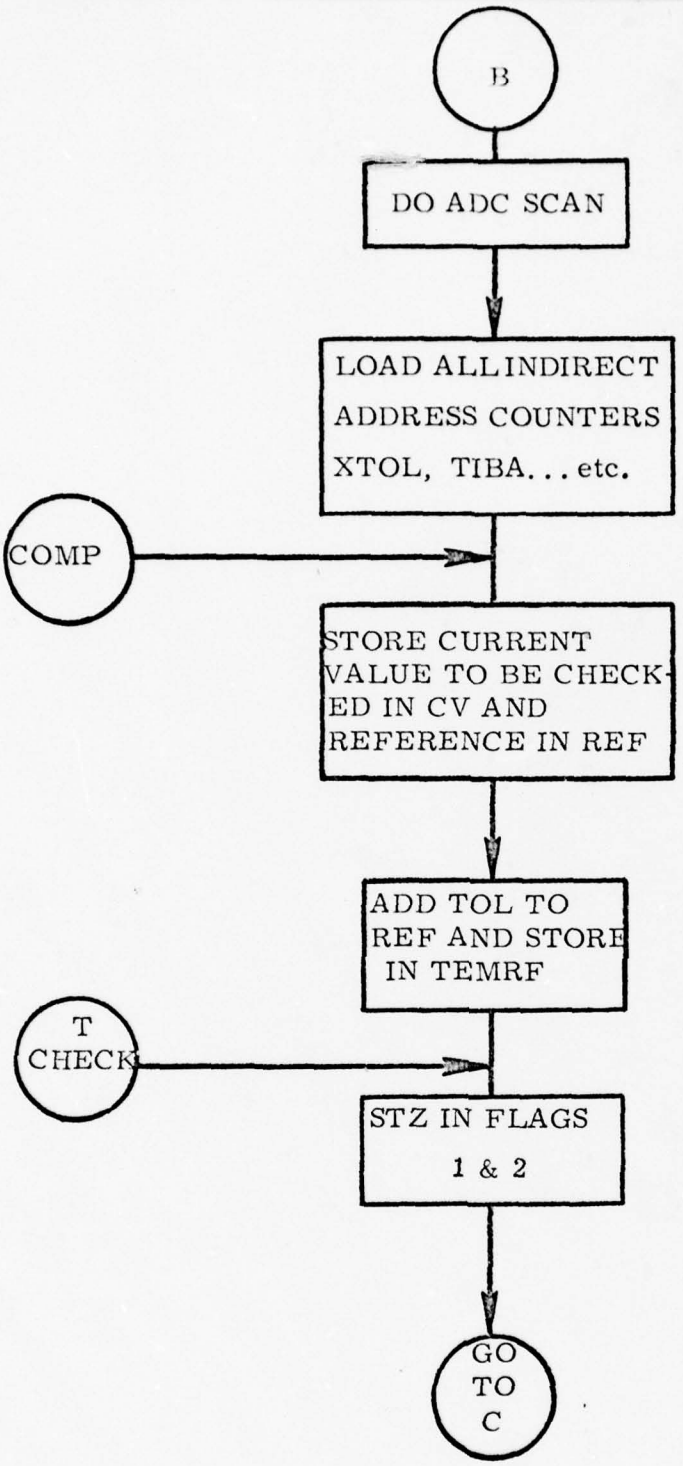
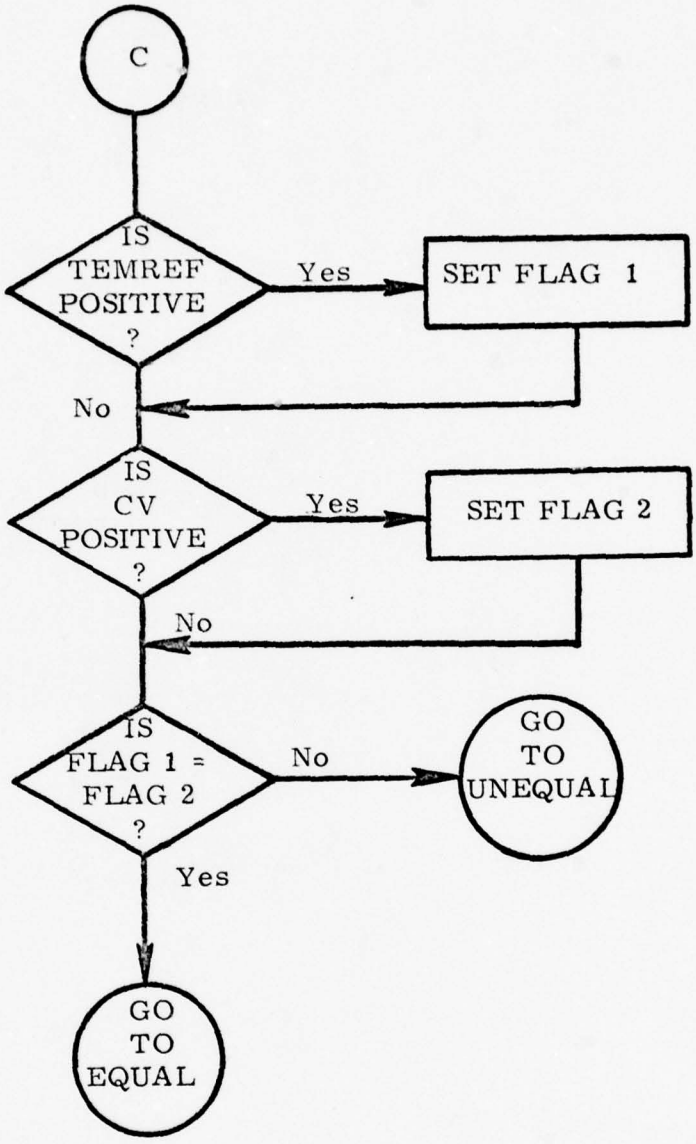


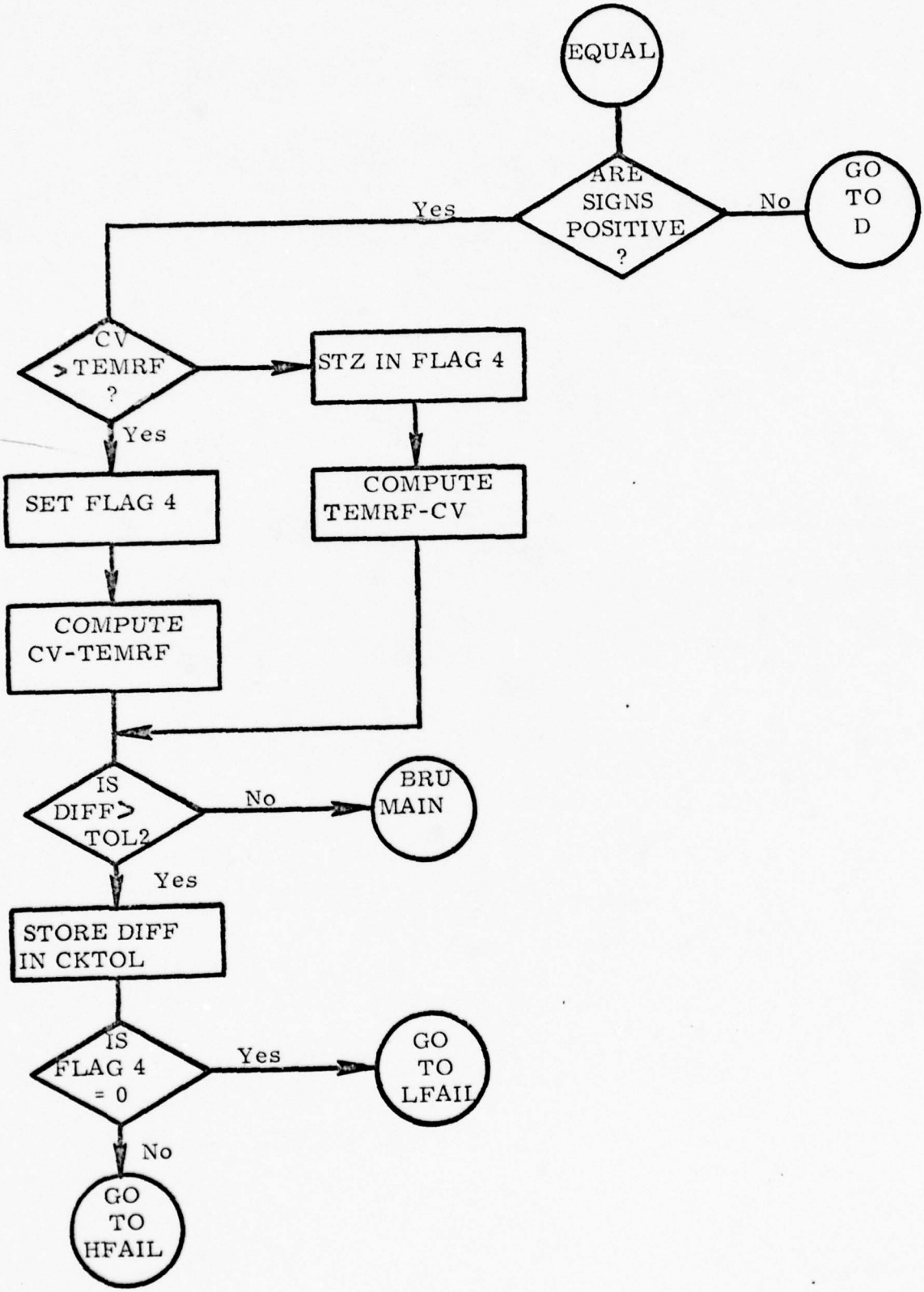
FIGURE 3.2 ADC CHECK PROGRAM FLOW CHART

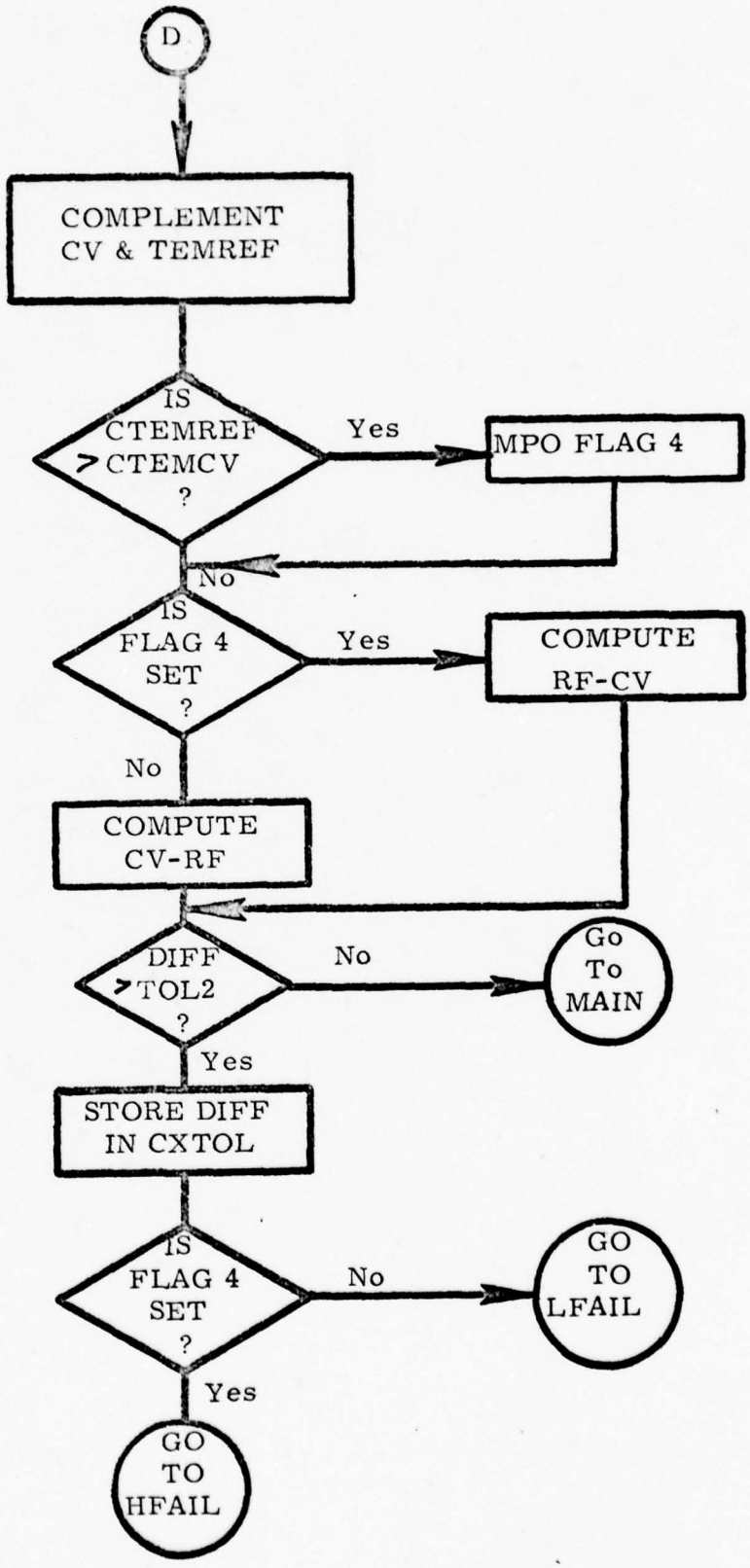

MAIN 2
COMPUTE 2 x TOL
STZ PRINT COUNT
ZERO OUT ALL
BUFFERS AND FLAGS
SS1 SET ?
No
GO TO PRINT
HALT
Yes
SS6 SET ?
Yes
HALT
No
GO TO A

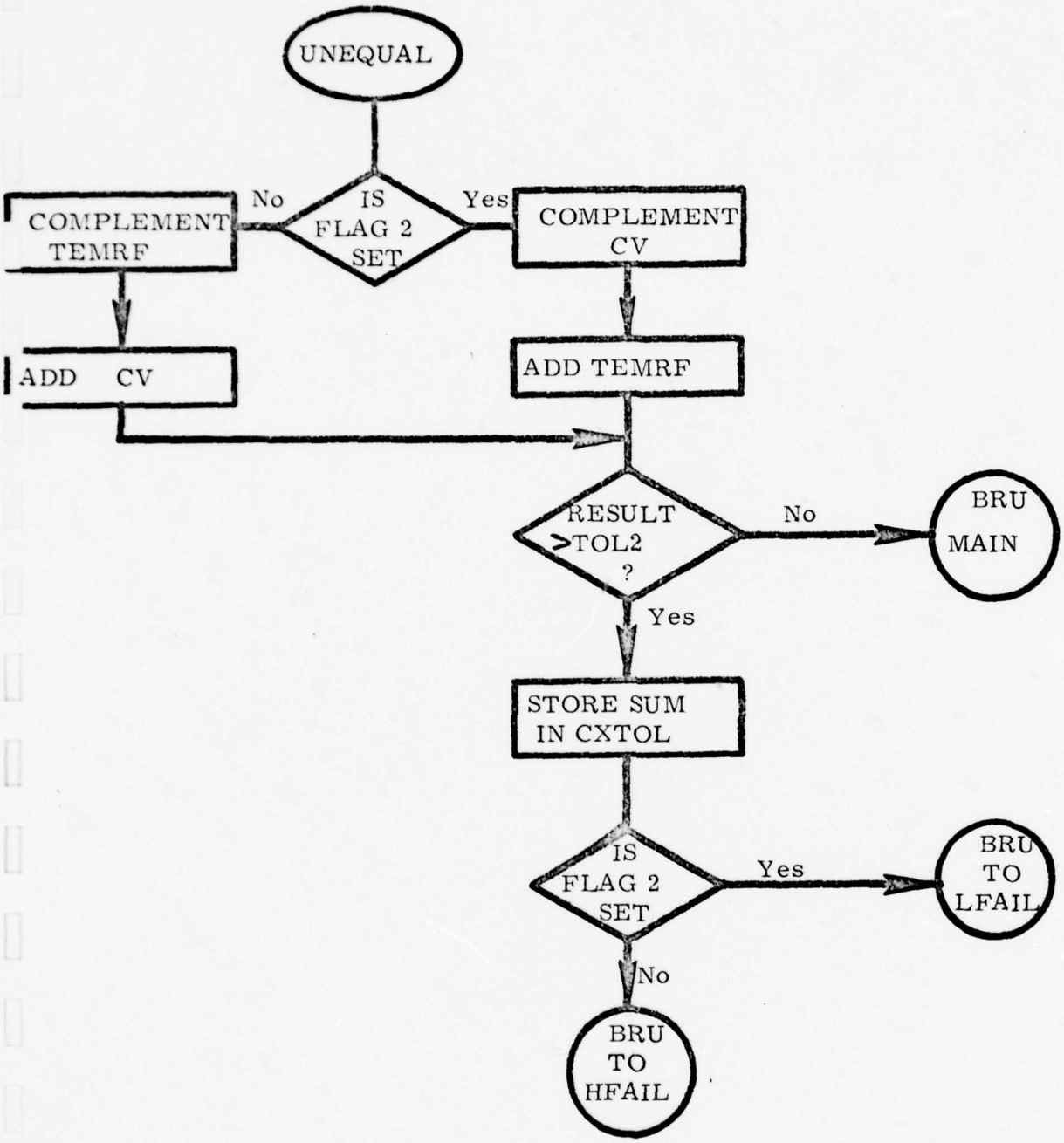

A
SS2 SET ?
Yes
STORE REF BUFFER FIRST AND LAST ADDRESS IN SRVA & LSRVA
No
STORE SCAN BUFFER FIRST AND LAST ADDRESS IN SRVA & LSRVA
IS SCAN FLAG HIGH ?
Yes
No
ZERO OUT INPUT BUFFER
DO ADC SCAN
MOVE SCAN VALUES TO SCAN REFERENCE BUFFER
ZERO OUT INPUT BUFFER
GO TO B

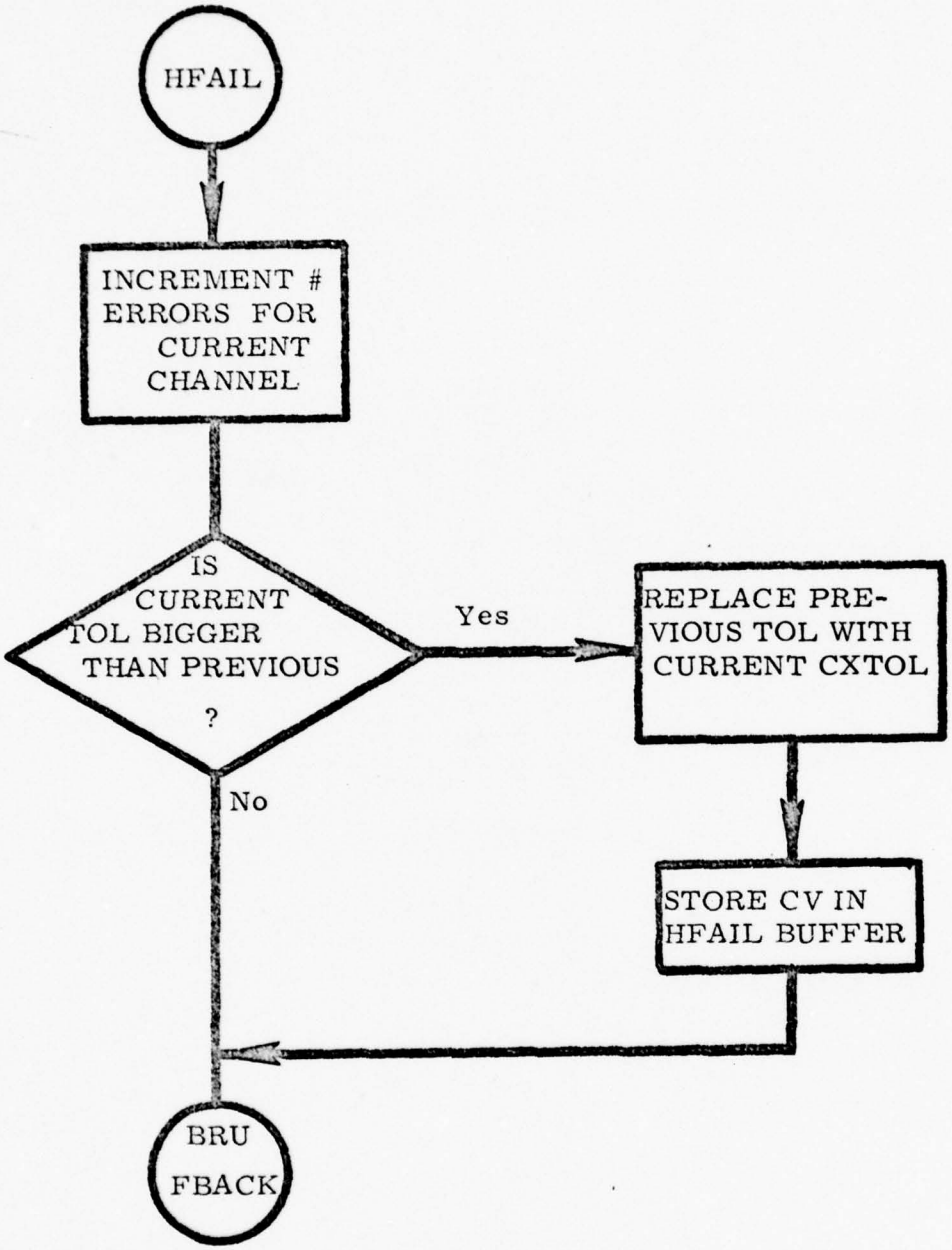

B
DO ADC SCAN
LOAD ALL INDIRECT ADDRESS COUNTERS XTOL, TIBA... etc.
COMP
STORE CURRENT VALUE TO BE CHECKED IN CV AND REFERENCE IN REF
ADD TOL TO REF AND STORE IN TEMRF
T CHECK
STZ IN FLAGS 1 & 2
GO TO C

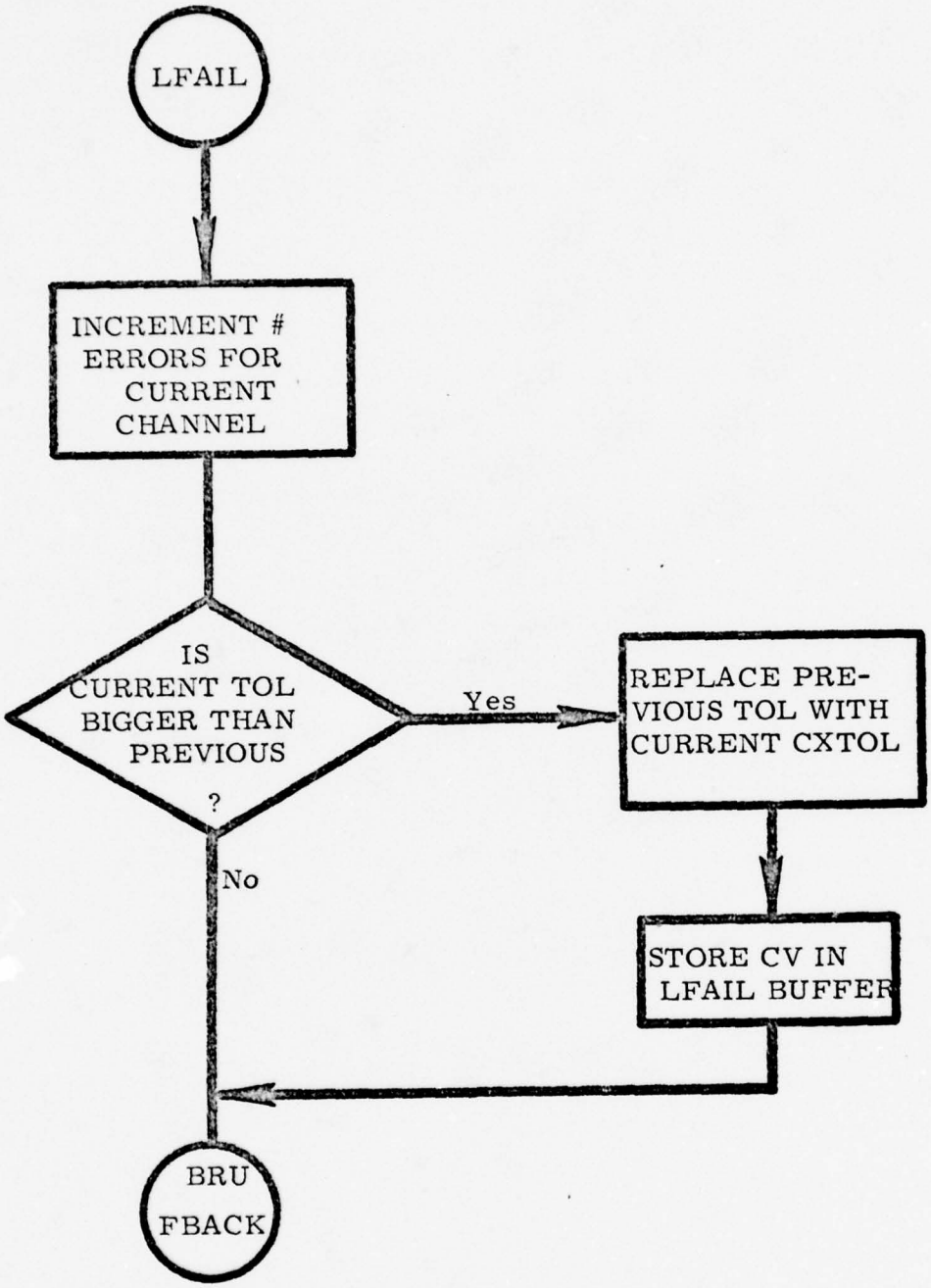

C
IS TEMREF POSITIVE ?
Yes
SET FLAG 1
No
IS CV POSITIVE ?
Yes
SET FLAG 2
No
IS FLAG 1 = FLAG 2 ?
No
GO TO UNEQUAL
Yes
GO TO EQUAL

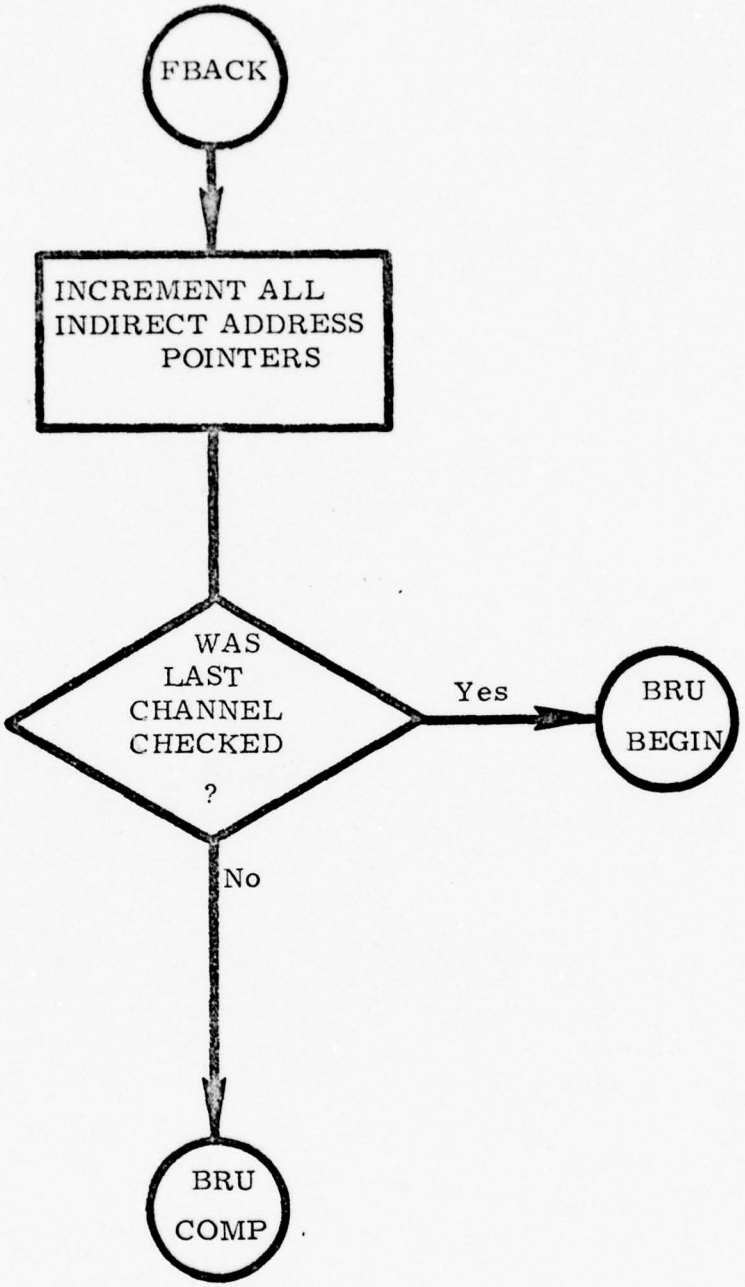

EQUAL
ARE SIGNS POSITIVE ?
Yes
No
GO TO D
CV > TEMRF ?
Yes
STZ IN FLAG 4
SET FLAG 4
COMPUTE TEMRF-CV
COMPUTE CV-TEMRF
IS DIFF> TOL2
No
BRU MAIN
Yes
STORE DIFF IN CKTOL
IS FLAG 4 = 0
Yes
GO TO LFAIL
No
GO TO HFAIL

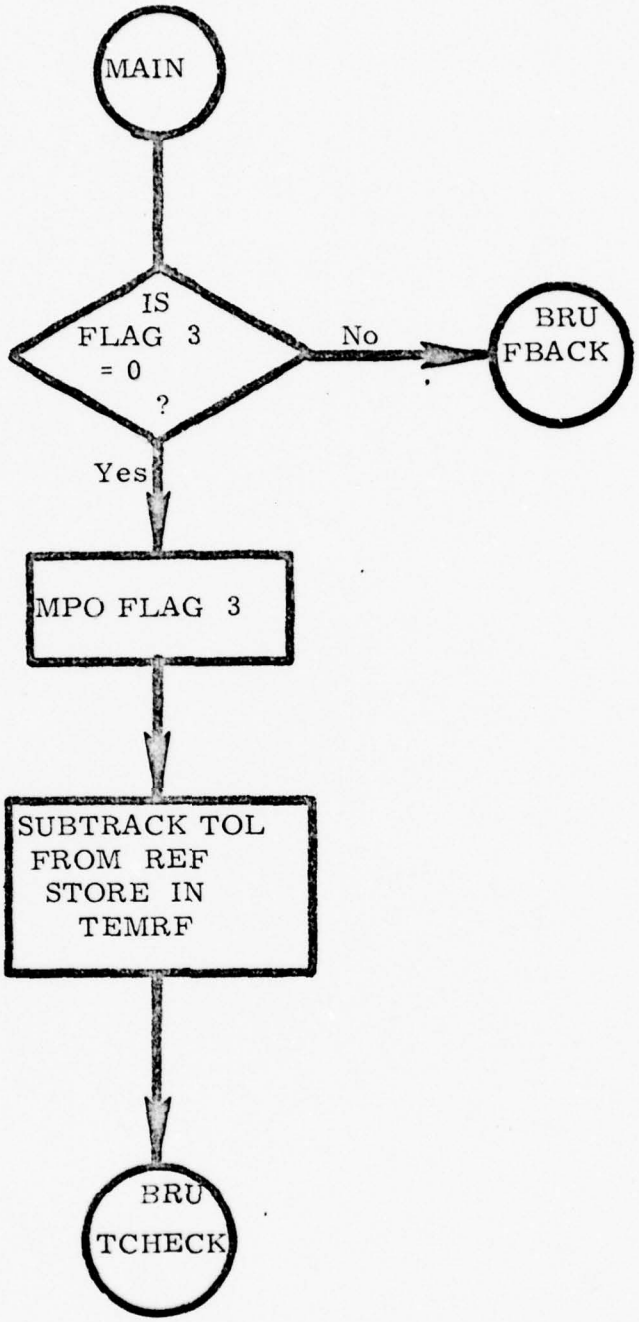

D
COMPLEMENT CV & TEMREF
IS CTEMREF > CTEMCV ?
Yes
MPO FLAG 4
No
IS FLAG 4 SET ?
Yes
COMPUTE RF-CV
No
COMPUTE CV-RF
DIFF > TOL2 ?
No
Go To MAIN
Yes
STORE DIFF IN CXTOL
IS FLAG 4 SET ?
No
GO TO LFAIL
Yes
GO TO HFAIL

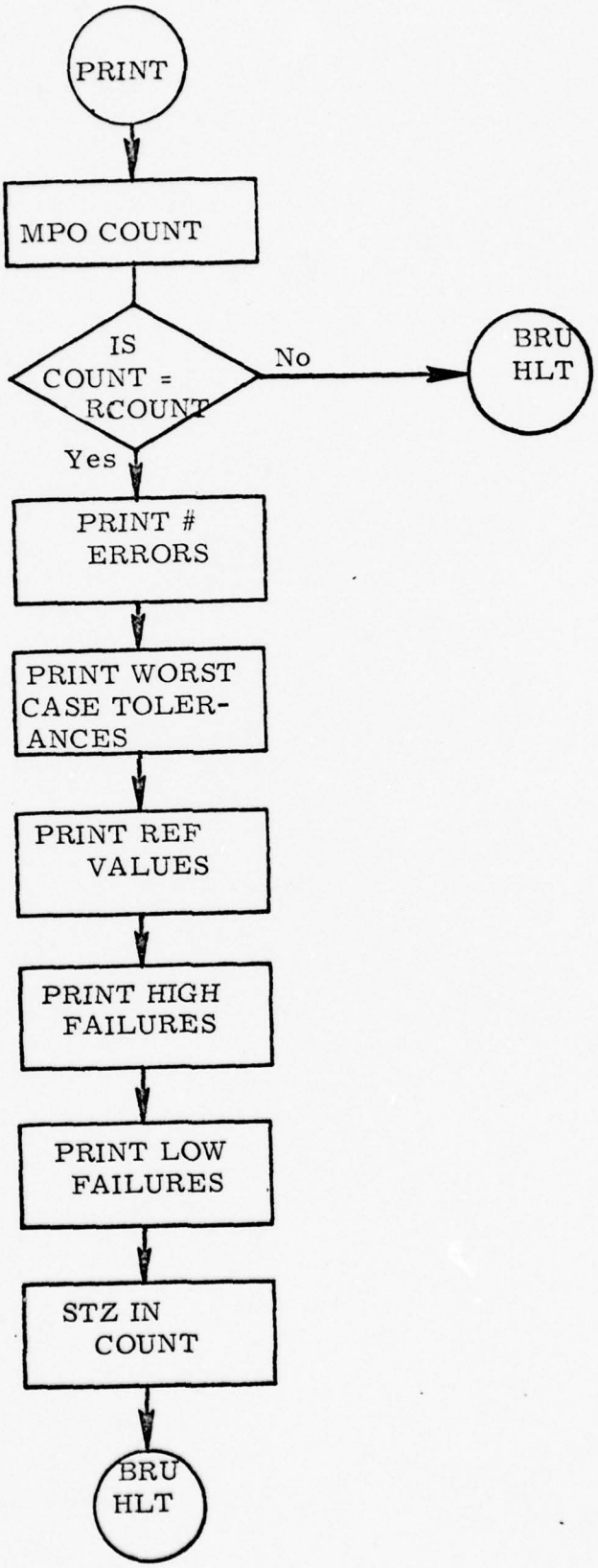

UNEQUAL
IS FLAG 2 SET
No
Yes
COMPLEMENT TEMRF
COMPLEMENT CV
ADD CV
ADD TEMRF
RESULT >TOL2 ?
No
BRU MAIN
Yes
STORE SUM IN CXTOL
IS FLAG 2 SET
Yes
BRU TO LFAIL
No
BRU TO HFAIL


HFAIL
INCREMENT # ERRORS FOR CURRENT CHANNEL
IS CURRENT TOL BIGGER THAN PREVIOUS ?
Yes
REPLACE PREVIOUS TOL WITH CURRENT CXTOL
No
STORE CV IN HFAIL BUFFER
BRU FBACK


LFAIL
INCREMENT # ERRORS FOR CURRENT CHANNEL
IS CURRENT TOL BIGGER THAN PREVIOUS ?
Yes
REPLACE PREVIOUS TOL WITH CURRENT CXTOL
No
STORE CV IN LFAIL BUFFER
BRU FBACK


FBACK
INCREMENT ALL INDIRECT ADDRESS POINTERS
WAS LAST CHANNEL CHECKED ?
Yes
BRU BEGIN
No
BRU COMP


MAIN
IS FLAG 3 = 0 ?
No
BRU FBACK
Yes
MPO FLAG 3
SUBTRACK TOL FROM REF STORE IN TEMRF
BRU TCHECK


PRINT
MPO COUNT
IS COUNT = RCOUNT
No
BRU HLT
Yes
PRINT # ERRORS
PRINT WORST CASE TOLER-ANCES
PRINT REF VALUES
PRINT HIGH FAILURES
PRINT LOW FAILURES
STZ IN COUNT
BRU HLT

APPENDIX A

CROSS REFERENCE OF REAL TIME CODE AND RELATED EQUATIONS

B-K DYNAMICS, INC.

COMPUTATION OF IRIS RATIO NO. 7

RLB = (ℓ/B) / Scalet

E111 = COS (ϵ)

RFEET = R_i

HR7E = $H\Big|_{\epsilon}^{R=7000}$ (interpolated from table HR7EE)

$$HC7FT = \left[\frac{R_i}{22965.831}\right]^{-2.341} \left[H\Big|_{\epsilon}^{R=7000} \right] \left(\frac{f^2}{T_{f,\,plume}}\right) = \frac{H_{c7} f^2}{T_{f,\,plume}}$$

$$P1IRSS = P_1 = 2\tan^{-1} \left(\frac{.5}{\ell/B}\right) = 2\tan^{-1} \left(\frac{.04884004884}{(\ell/B) \;/\; Scalet}\right)$$

Note: .04884004884 = .5/10.2375 and B = 3.0

P2IRSS = P_2 = 22.5 (10.2375) ((ℓ/B) /10.2375) / R (inches)

$$AT7 = \frac{P_2^2 \sin (P_1)}{2 \cos^2 (P_1^2/2)} = A_{t7}$$

$$PJTU1 = 0.1\, A_{t7}^{1.003258} = 0.1 e^{1.003258 \ln A_{t7}} = J_{tu1,\,plume}$$

$$FI7 = \frac{H_{c7} f^2}{T_{f,\,plume}}$$

$$RN = \left(\frac{H_{c7} f^2}{T_{f,\,plume}}\right) \;/\; J_{tu1,\,plume}$$

COMPUTATION OF SCALED PLUME ROTATION ANGLE

$$F1 = L_x$$

$$F2 = L_y$$

$$F3 = L_z$$

$$G1 = (\cos(\alpha) + \sin(\alpha)\, \dot{Z}_{TE} / \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2})\, \dot{X}_{TE}$$

$$G2 = (\cos(\alpha) + \sin(\alpha)\, Z_{TE} / \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2})\, \dot{Y}_{TE}$$

$$G3 = \dot{Z}_{TE} \cos(\alpha) - \sin(\alpha) \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}$$

$$VTI = V_T$$

$$RC1 = \sqrt{L_x^2 + L_y^2}$$

$$RC1 = L_y / \sqrt{L_x^2 + L_y^2} = \cos(\beta)$$

$$RCB = \cos^{-1}(\beta) = \beta$$

$$RCX = -\cos(\beta) = C_x$$

$$RCY = -\sin(\beta) = C_y$$

$$IF\ (L_x \geq 0),\ C_y = \sin(\beta)$$

$$F11 = (L_y (\dot{Z}_{TE} \cos(\alpha) - \sin(\alpha) \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2})$$

$$-L_z (\dot{Y}_{TE} \cos(\alpha) + \sin(\alpha)\, \dot{Y}_{TE}\, \dot{Z}_{TE} / \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2})) / V_T$$

$$F22 = (L_z (\cos(\alpha) + \sin(\alpha)\, \dot{Z}_{TE} / \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2})\, \dot{X}_{TE}$$

$$-L_x (\dot{Z}_{TE} \cos(\alpha) - \sin(\alpha) \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}\,)) / V_T$$

$$F33 = (L_x (\cos(\alpha) + \sin(\alpha)\, \dot{Z}_{TE} / \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2})\, \dot{Y}_{TE}$$

$$-L_y (\cos(\alpha) + \sin(\alpha)\, \dot{Z}_{TE} / \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2})\, \dot{X}_{TE}) / V_T$$

Note: $$\bar{C}_L = i_E \dot{X}_{TE} (\cos(\alpha) + \dot{Z}_{TE} \sin(\alpha) / \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}) / V_T$$

$$+j_E \dot{Y}_{TE} (\cos(\alpha) + \dot{Z}_{TE} \sin(\alpha) / \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}) / V_T$$

$$+k_E (\dot{Z}_{TE} \cos(\alpha) - \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2} \sin(\alpha)) / V_T$$

Therefore: $$F11 = (L_y C_{Lz} - L_z C_{Ly})$$

$$F22 = (L_z C_{Lx} - L_x C_{Lz})$$

$$F33 = (L_x C_{Ly} - L_y C_{Lx})$$

$$FCR = \sqrt{(L_y C_{Lz} - L_z C_{Ly})^2 + (L_z C_{Lx} - L_x C_{Lz})^2 + (L_x C_{Ly} - L_y C_{Lx})^2}$$

$$G11 = (L_y C_{Lz} - L_z C_{Ly})\, C_x$$

$$G22 = (L_z C_{Lx} - L_x C_{Lz})\, C_y$$

$$C111 = \frac{(L_y C_{Lz} - L_z C_{Ly})\, C_x + (L_z C_{Lx} - L_x C_{Lz})\, C_y}{\sqrt{(L_y C_{Lz} - L_z C_{Ly})^2 + (L_z C_{Lx} - L_x C_{Lz})^2 + (L_x C_{Ly} - L_y C_{Lx})^2}}$$

$$= \cos(\theta)$$

$T111 = \cos^{-1}(\cos\theta) = \theta$

IF $(\bar{L}_{os} \times \overline{C_L})$ ZE - COMP ≥ 0, $T_{RP} = \pi/2 - \theta$

IF $(\bar{L}_{os} \times \overline{C_L})$ ZE - COMP < 0 and $\theta < \pi/2$, $T_{RP} = \theta + \pi/2$

IF $(\bar{L}_{os} \times \overline{C_L})$ ZE - COMP < 0 and $\theta \geq \pi/2$, $T_{RP} = \theta - 3\pi/2$

SPO = T_{rP} / Scale Factor

COMPUTATION OF , COS (ϵ) AND R/ℓ

SKK = K

S2 = cos θ_L cos ψ_L

SPL = sin ψ_L

S3 = sin θ_L cos ψ_L

S4 = cos θ_L sin ψ_L

CPL = cos ψ_L

S5 = sin θ_L sin ψ_L

CTL = cos θ_L

VTI = VT

$A1 = X_E^2 + Y_E^2 + Z_E^2$

$A1 = (.01745329)(4.637084242)\sqrt{A1} / (\ VT^2 C_{LA})$

SA = sin (α)

CA = cos (α)

$XC = \dot{X}_{TE}$ (200) (102.375) /K

$$YC = \dot{Y}_{TE} (200) (102.375) /K$$

$$ZC = \dot{Z}_{TE} (200) (102.375) /K$$

$$XTA = XE$$

$$YTA = YE$$

$$ZTA = ZE$$

} interpolated values

$$XXX = X_G$$

$$YYY = Y_G$$

$$ZZZ = Z_G$$

$$RRR = X_G^2 + Y_G^2 + Z_G^2$$

$$RRR = V_T \sqrt{X_G^2 + Y_G^2 + Z_G^2}$$

$$F1 = X_G \cos \theta_L \cos \psi_L - Y_G \sin \psi_L + Z_G \sin \theta_L \cos \psi_L$$

$$F2 = X_G \cos \theta_L \sin \psi_L + Y_G \cos \psi_L + Z_G \sin \theta_L \sin \psi_L$$

$$F3 = Z_G \cos \theta_L - X_G \sin \theta_L$$

$$SR = \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}$$

$$S1 = \cos(\alpha) + \sin(\alpha)\,\dot{Z}_{TE} \Big/ \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}$$

$$G1 = \dot{X}_{TE}\cos(\alpha) + \sin(\alpha)\,\dot{X}_{TE}\dot{Z}_{TE} \Big/ \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}$$

$$G2 = \dot{Y}_{TE}\cos(\alpha) + \sin(\alpha)\,\dot{Y}_{TE}\dot{Z}_{TE} \Big/ \sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}$$

$$G3 = \dot{Z}_{TE}\cos(\alpha) - \sin(\alpha)\sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}$$

$$E111 = \Bigg\{ (X_G \cos\theta_L \cos\psi_L - Y_G \sin\psi_L + Z_G \sin\theta_L \cos\psi_L)$$

$$\frac{(X_{TE}\cos(\alpha) + \sin(\alpha)\,\dot{X}_{TE}\dot{Z}_{TE})}{\sqrt{\dot{X}_{TE} + \dot{Y}_{TE}}}$$

$$+ (X_G \cos\theta_L \sin\psi_L + Y_G \cos\psi_L + Z_G \sin\theta_L \sin\psi_L)$$

$$\frac{(\dot{Y}_{TE}\cos(\alpha) + \sin(\alpha)\,\dot{Y}_{TE}\dot{Z}_{TE})}{\sqrt{\dot{X}_{TE} + \dot{Y}_{TE}}}$$

$$+ (Z_G \cos\theta_L - X_G \sin\theta_L)(\dot{Z}_{TE}\cos(\alpha) - \sin(\alpha)\sqrt{\dot{X}_{TE}^2 + \dot{Y}_{TE}^2}) \Bigg\}$$

$$/\, V_T\,(X_G^2 + Y_G^2 + Z_G^2) = \cos(\epsilon)$$

$$COSE = 1.0 - E111^2$$

$$RLB = 1.0 - E111^2$$

$$RLB = (\sqrt{1.0 - E111^2})\,(RLBK) \,/\, \text{Scale Factor}$$

$$COSE = E111 \,/\, 1.02375$$

$$\bar{A}_{AERO} = \bar{i}_E \ddot{X}_{TE} + \bar{j}_E \ddot{Y}_{TE} + \bar{k}_E (\ddot{Z}_{TE} - g)$$
$$\bar{a}_I = \bar{i}_E \ddot{X}_{TE} + \bar{j}_E \ddot{Y}_{TE} + \bar{k}_E \ddot{Z}_{TE}$$

SUBSCRIPTS:

- E – EARTH FIXED COORDINATES
- GN – GENERALIZED TARGET COORDINATES
- F – TARGET FIXED COORDINATES
- L – LAUNCH COORDINATES

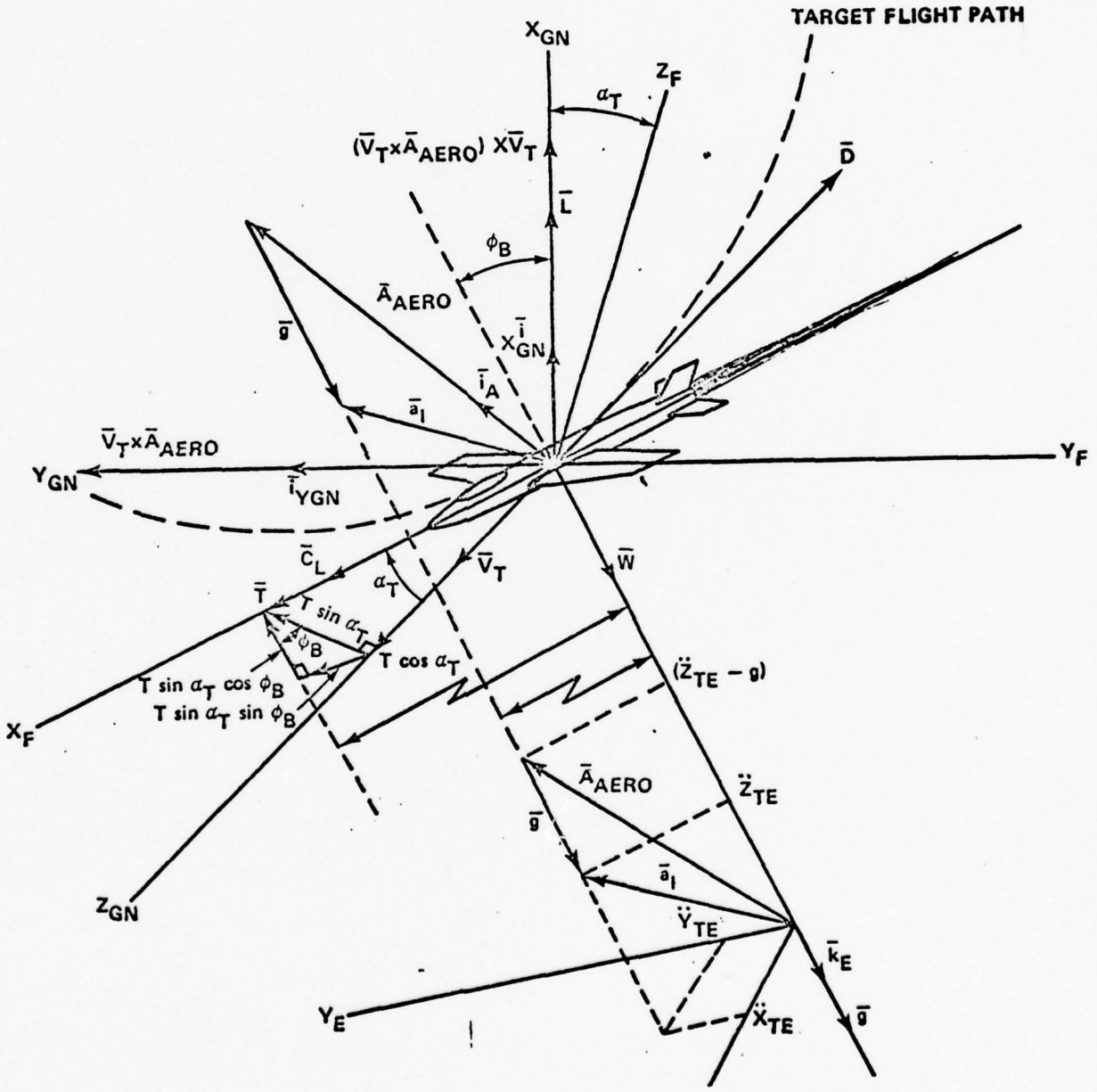


FIGURE A. 1

TARGET FIXED COORDINATE SYSTEM

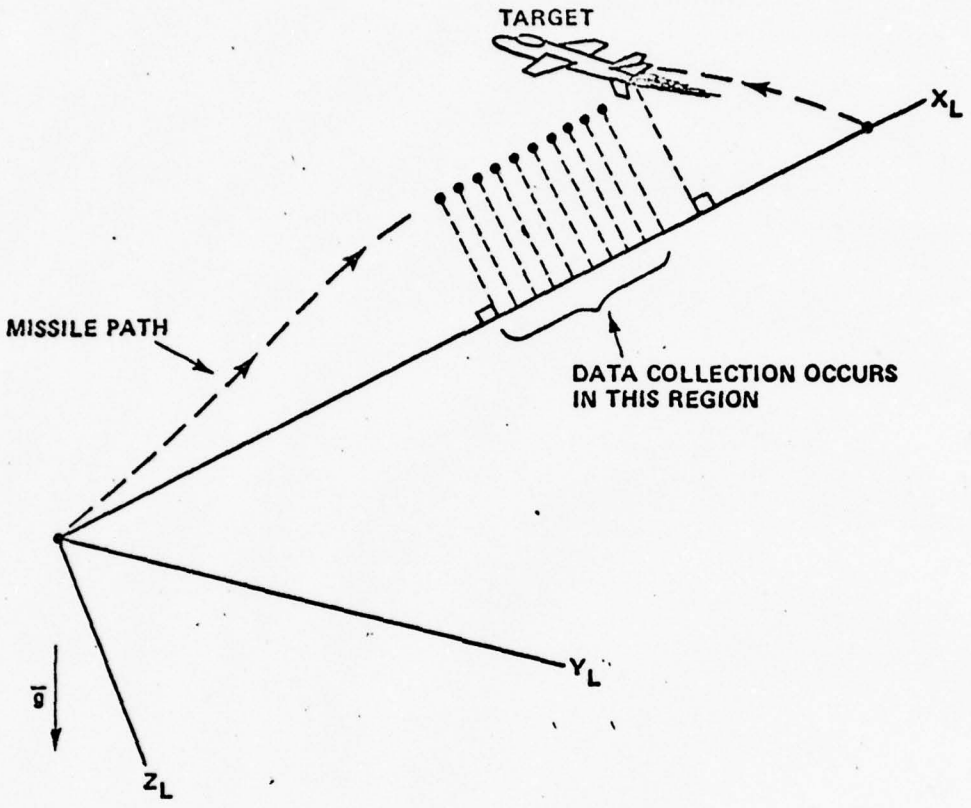


FIGURE A.2

RANGE-TO-GO

CONTROL OF A/D DATA COLLECTION

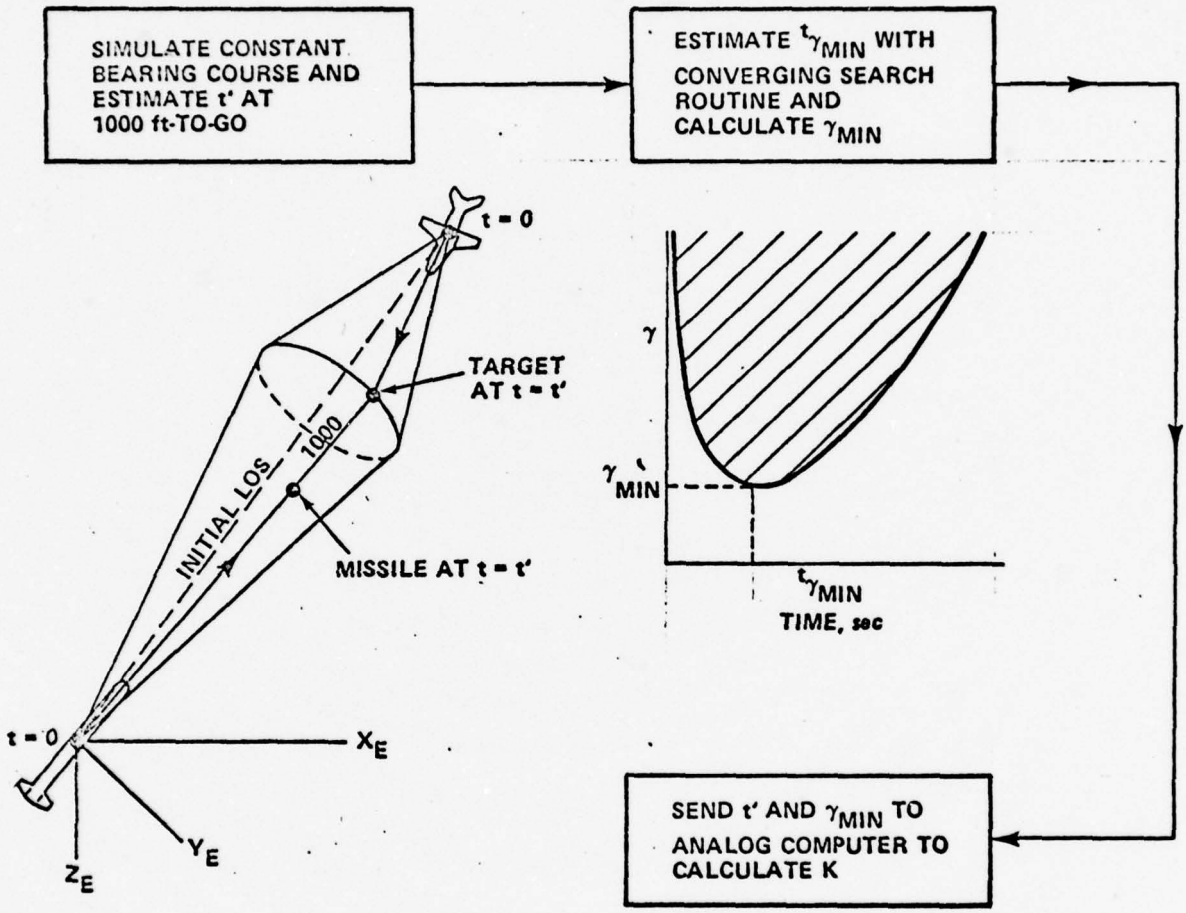


FIGURE A.3

SEQUENCE OF VARIABLE SCALE FACTOR CALCULATION

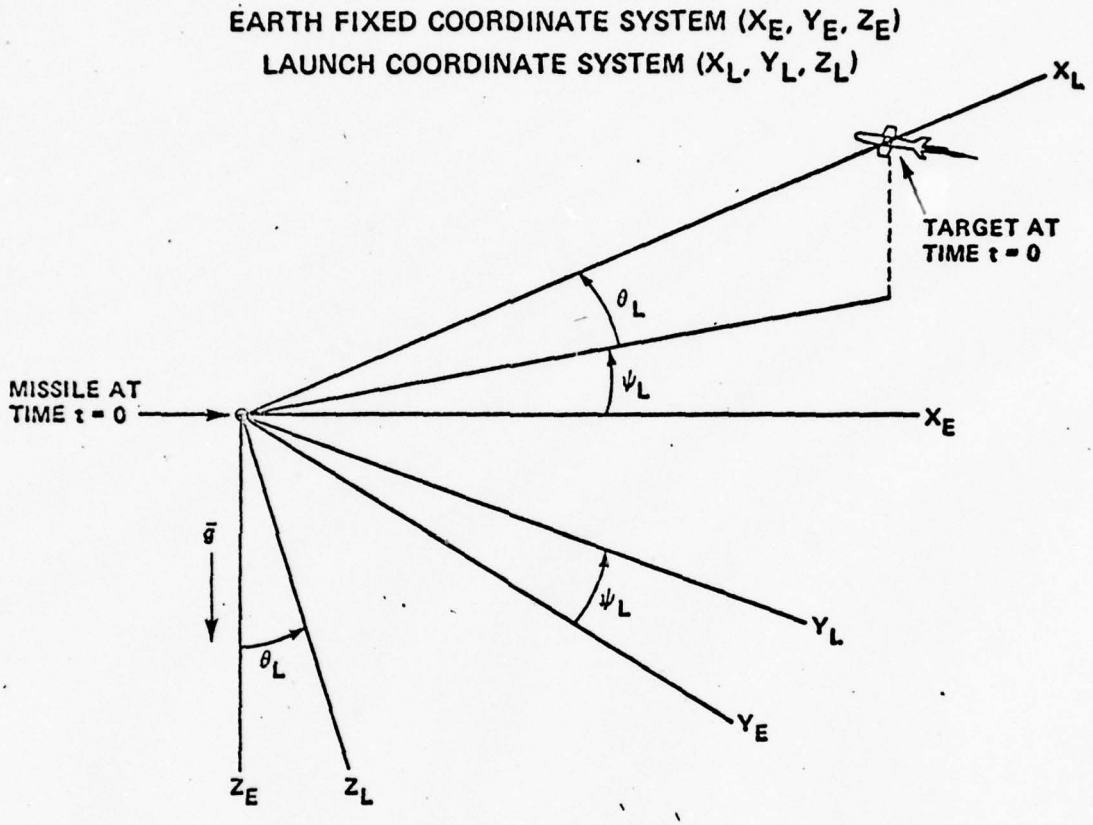


FIGURE A.4

EARTH FIXED AND LAUNCH COORDINATE SYSTEMS

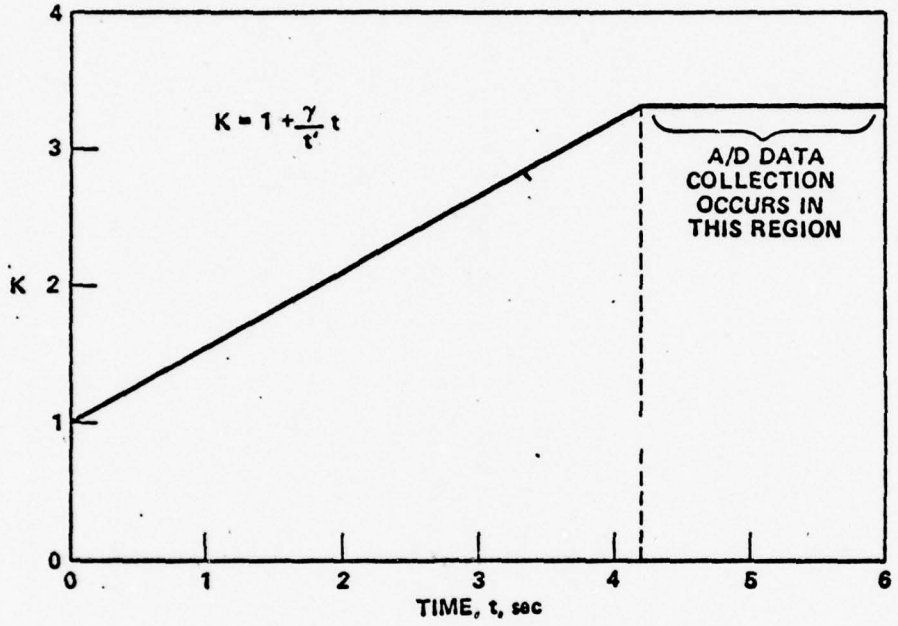


FIGURE A.5

VARIABLE ANALOG SCALE FACTOR

APPENDIX B

AD/4 TRUNK LINE DISCRETE PATCHING

BOK
DYNAMICS, INC.

00	10	20	30	40	50	60	70
01	11	21	31	41	51	61	71
02	12	22	32	42	52	62	72
03	13	23	33	43	53	63	73
04	14	24	34	44	54	64	74
05	15	25	35	45	55	65	75
06	16	26	36	46	56	66	76
07	17	27	37	47	57	67	77
I	O	I	O	I	O	I	O

VARIABLE 1 VARIABLE 2

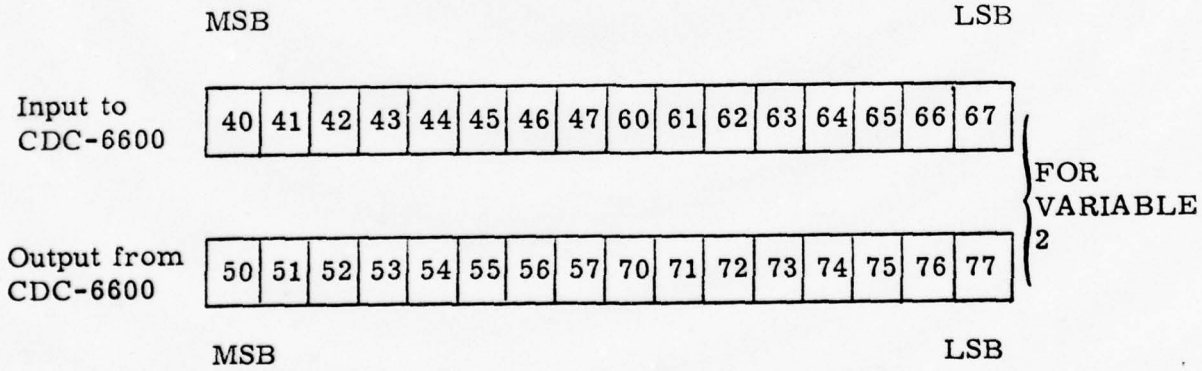


FIGURE B. 1 AD/4 TRUNK LINE DISCRETES

(FOR AD/4 # 1)

00	10	20	30	40	50	60	70
01	11	21	31	41	51	61	71
02	12	22	32	42	52	62	72
03	13	23	33	43	53	63	73
04	14	24	34	44	54	64	74
05	15	25	35	45	55	65	75
06	16	26	36	46	56	66	76
07	17	27	37	47	57	67	77
		I	O	I	O		

VARIABLE 1

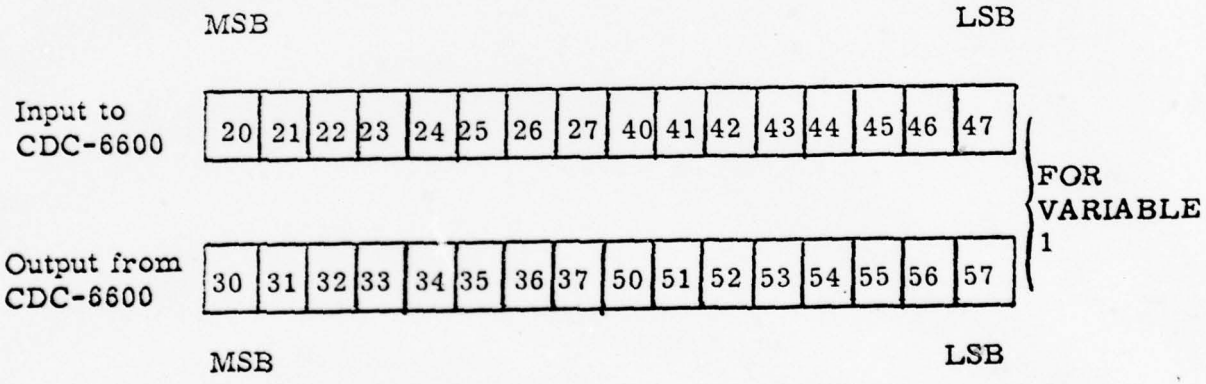


FIGURE B.1 AD/4 TRUNK LINE DISCRETES
(FOR AD/4 # 2)

TABLE B.1
SUMMARY OF POSSIBLE DISCRETE TRUNK LINE CONNECTIONS

CDC-6600 INPUT DISCRETES

W50 -- INPUT DISCRETE 1 -- COMMON /*IDIS2/IIN
W51 -- INPUT DISCRETE 2 -- COMMON /*IDIS2/2,IIN
W52 -- INPUT DISCRETE 3 -- COMMON /*IDIS2/3,IIN
W53 -- INPUT DISCRETE 4 -- COMMON /*IDIS2/4,IIN

POSSIBLE AD/4 CONNECTIONS

V50 -- AD/4 #1 LOGIC TRUNKS 00-07 AND 20-27
V52 -- AD/4 #1 LOGIC TRUNKS 40-47 AND 60-67
V60 -- AD/4 #2 LOGIC TRUNKS 20-27 AND 40-47

CDC-6600 OUTPUT DISCRETES

W60 -- OUTPUT DISCRETE 1 -- COMMON /*ODIS2/IOUT
W61 -- OUTPUT DISCRETE 2 -- COMMON /*ODIS2/2,IOUT
W62 -- OUTPUT DISCRETE 3 -- COMMON /*ODIS2/3,IOUT
W63 -- OUTPUT DISCRETE 4 -- COMMON /*ODIS2/4,IOUT

POSSIBLE AD/4 CONNECTIONS

V51 -- AD/4 #1 LOGIC TRUNKS 10-17 AND 30-37
V53 -- AD/4 #1 LOGIC TRUNKS 50-57 AND 70-77
V61 -- AD/4 #2 LOGIC TRUNKS 30-37 AND 50-57

TABLE B.2
SUMMARY OF POSSIBLE ANALOG TRUNK LINE CONNECTIONS

CDC-6600 ADCS

W00 -- ADC GROUP 1 -- COMMON /*ADC1/ADIN(16)
W01 -- ADC GROUP 2 -- COMMON /*ADC1/17, ADIN(16)
W02 -- ADC GROUP 3 -- COMMON /*ADC1/33, ADIN(16)
W03 -- ADC GROUP 4 -- COMMON /*ADC1/49, ADIN(16)

CDC-6600 DACS

W10 -- DAC GROUP 1 -- COMMON /*DAC1/DAOUT(16)
W11 -- DAC GROUP 2 -- COMMON/* DAC1/17, DAOUT(16)
W12 -- DAC GROUP 3 -- COMMON/*DAC1/33, DAOUT(16)
W13 -- DAC GROUP 4 -- COMMON/*DAC1/49, DAOUT(16)

AD/4 ANALOG CONNECTIONS (ADCS OR DACS)

V00 -- AD/4 #1 FIELD 0 ANALOG TRUNKS 10-17 AND 30-37
V01 -- AD/4 #1 FIELD 0 ANALOG TRUNKS 50-57 AND 70-77
V02 -- AD/4 #1 FIELD 1 ANALOG TRUNKS 10-17 AND 30-37
V03 -- AD/4 #1 FIELD 1 ANALOG TRUNKS 50-57 AND 70-77
V04 -- AD/4 #1 FIELD 2 ANALOG TRUNKS 10-17 AND 30-37
V05 -- AD/4 #1 FIELD 2 ANALOG TRUNKS 50-57 AND 70-77
V06 -- AD/4 #1 FIELD 3 ANALOG TRUNKS 10-17 AND 30-37
V07 -- AD/4 #1 FIELD 3 ANALOG TRUNKS 50-57 AND 70-77
U07 -- AD/4 #2 FIELD 1 ANALOG TRUNKS 50-57 AND 70-77
U06 -- AD/4 #2 FIELD 2 ANALOG TRUNKS 50-57 AND 70-77

APPENDIX C

COMPUTER CODE FOR PRODUCING *NORMALIZED RANDOM NUMBERS*

B✪K
DYNAMICS, INC.

```
SUBROUTINE NORMAL ( RX,XL,XU,MU,SIGMA)

DESCRIPTION
    THIS ROUTINE PRODUCES NORMALIZED RANDOM NUMBERS WITHIN
    A RANGE SPECIFIED BY THE USER.

INPUT
    1  XL       LOWER LIMIT OF NORMAL CURVE
    2  XU       UPPER LIMIT OF NORMAL CURVE
    3  MU       MEAN OF NORMAL CURVE (TYPE REAL)
    4  SIGMA    STANDARD DEVIATION OF NORMAL CURVE

OUTPUT
    1  RX       THE NORMALIZED RANDOM NUMBER
```

REF: General Purpose Computer Subroutines
Report No. TR-WS-75-2
USAMC, Redstone Arsenal, Alabama
January 1975

```
      SUBROUTINE NORMAL( RX, XL, XU, MU, SIGMA )
      DIMENSION FZ(126)
      REAL MU
      DATA DT,DT1,NTP,NPT,SQ2PI/.01,.04,126.0,.3989422803/
      SLS = MU - 5.0 * SIGMA
      SUS = MU + 5.0 * SIGMA
      IF( XL .LT. SUS .AND. XU .GT. SLS )  GO TO 10
      PRINT 901, SLS, SUS, XL, XU, MU, SIGMA
  901 FORMAT(*)LIMITS FOR NORMAL DISTRIBUTION SHOULD BE BET*,F10.2
     1       * AND*F10.2 * --- PROGRAM TERMINATED.*///* XL =*F10.2,5X
     2       * XU =* F10.2,5X* MU =*F10.2,5X* SIGMA =*F10.2)
      STOP 123
   10 IF(NPT.NE.0) GO TO 2
      T2=DT1
      FZ(1)=.5
      FC=FZ(1)
      T=0.
      FP=SQ2PI*EXP(-.5*T*T)
      DT2=DT*.5
      NPT=1
    1 T = T+DT
      F = SQ2PI*EXP(-.5*T*T)
      FC=FC + DT2*(F+FP)
      FP=F
      IF(ABS(T-T2).GT..0001) GO TO 1
      NPT=NPT + 1
      FZ(NPT)=FC
      T2=T2+DT1
      IF(NPT.LT.NTP) GO TO 1
    2 CONTINUE
      RN = RANF(R)
      R=RN
      IF(RN.LT..5) R=1.-RN
      IF(R.GT.FZ(NPT)) GO TO 2
      DO 3 I=1,NPT
      IF(R.GT.FZ(I)) GO TO 3
      IF(I.EQ.1) GO TO 4
      X=I-1
      X=X*DT1
      X1=X-DT1
      RX = X1 +(R-FZ(I-1))*DT1/(FZ(I)-FZ(I-1))
      GO TO 5
    3 CONTINUE
      PRINT 100
  100 FORMAT(13H0 ERROR DISRN)
      STOP
    4 CONTINUE
      RX=0.
    5 CONTINUE
      IF(RN.LT..5) RX=-RX
      RX= SIGMA*RX+MU
      IF(RX.GT.XU) GO TO 2
      IF(RX.LT.XL) GO TO 2
      RETURN
      END
```

APPENDIX D

STINGER SKELETON REAL TIME CHECKOUT PROGRAM

BOK DYNAMICS, INC.

SKELETON REAL TIME PROGRAM

EXECUTION PROCEDURE:

NOTE: The B-K microprocessor is to be programmed and connected to the appropriate trunks on the AD/4 logic board as specified in the section "MICROPROCESSOR REAL TIME SET-UP" to handle the discrete communication with the digital program. The symbol Δ means the "SEND" key on the DDS terminal.

1. Make sure all the input discretes from the microprocessor are low.

2. Catalog the SKELETON REAL TIME program by submitting the deck with the following cards in front:

```
XXXGS, T5, CM6000.
ACCT(PN=XXXXXXX, PBC=7364G13R00, CC=7300, OP=A3, JN=XXXX)
REQUEST, SAVE, *PF.
COPY, INPUT, SAVE.
CATALOG, SAVE, XXXGS, ID=G1XXXX.
EOR
```

(NOTE: X's represent characters to be supplied by the user.)

3. Sign on the DDS terminal as follows:

   Screen says "ACTIVE AND NOT BUSY"

   User types Δ

   Screen says "TYPE USER NM, PASSWORD, PBC, CC, PRG NM, JOB NO"

   User types XXXXXXX, XXXXXXX, 7364G13R00, 7300, XXXXXXX, XXXX Δ

Screen says "SELECT A COMMAND CODE FROM THE LIST BELOW"

O Operator Aid
J Job Control
S File Scan
G File Generation and Updating
U File Utilities
P Permanent Files
T Terminate

4. User types P △ to get permanent file command set.

5. User types ATTACH(SKELRT, XXXGS, ID=G1XXXX) △

   where the XXXGS and G1XXXX are the same as on the catalog card in Step 2.

   Screen says "FUNCTION SUCCESSFUL", etc.

6. User types $J △ to get job control command set.

7. User types SKELRT, nn△ where nn is the first listed free control point (01, 02, etc.).

8. Screen says "SELECT STEP MODE IF WANTED", etc.

   User types △ to select automatic mode. Program will compile, load, and begin execution.

9. Screen says "PAUSE 1" with "PAUSED" on right side. At this time, all output discretes should be high. This is detected by the microprocessor as event #1.

10. User types GO △ (a second GO △ may be necessary). This will clear PAUSE 1.

    Screen says "PAUSE 2" with "PAUSED" on right side. At this time, all output discretes should be low--detected by the microprocessor as event #2.

11. Set input discrete 5 high using a switch on the microprocessor. User types GO △ (a second GO △ may be necessary). "PAUSE 2" will stay on screen but "PAUSED" on right side should disappear. Output discrete 4 should come high now--detected by microprocessor as event #3.

12. After a delay of approximately 15 seconds, the microprocessor will automatically set input discrete 6 high (event #4). This initiates the ADC-DAC conversion loop in the program.

13. When all conversions are complete, the program will set output discrete 7 high (microprocessor event #5) and terminate.

    Screen says "NO MORE CONTROL CARDS".

14. User types $U Δ to get file utility command set.

15. User types 1 Δ to terminate and save output file.

16. User types L, file name, XX, AU Δ in order to list the output file, where the file name is as listed on the bottom of the screen--RA1GSXX.

17. User types $T Δ and then Δ to sign off DDS.

MICROPROCESSOR REAL TIME SET-UP

The software checkout chassis's microprocessor is loaded with the program below using the front panel switches. The procedure is as follows:

- Select JAM and STEP mode (switches up)
- Load 000_8 into high and low address registers using LAH and LAL with 000_8 set on switch register
- Load program using DEPOSIT switch and switch register
- When program is loaded set 005_8 as switch register, select RUN and RUN on mode switches, hit INTERRUPT
- Program will now execute
- At completion of execution, hit INTERRUPT to re-execute

The AD/4 must be cabled to the processor as specified below:

AD/4 TR	20	line 8*	Checkout chassis DO	X**
	21	7		6
	22	6	Connect to switch on	CC
	23	5		X
	24	4		X
	25	3		X
	26	2		X
	27	1		X

AD/4 TR (SP)	0	line 16	Checkout chassis DI	7
	1			5
	2			1
	3			4
	4			8
	5			6
	6			3
	7			2

* This cable is installed between the AD/4 and software checkout chassis

** X Denotes "don't care"

MICROPROCESSOR REAL TIME PROGRAM

LOCATION	OP CODE	OPERATOR	CONNECTS
000	006	MVI A	
001	000	000	
002	121	A → P0	
003	123	A → P1	
004	101	0 → A	
005	074	CPI	
006	377	377	Check for all ones
007	110	JPZ	
010	004		
011	000		
012	006	MVI A	
013	001	01	Display event 1
014	121	A → P0	
015	101	0 → A	
016	074	CPI	
017	000	ZERO	Check for all zeroes
020	110	JNZ	
021	015		
022	000		
023	006	MVI A	
024	002	002	Display event 2
025	121	A → P0	
026	101	IP0 A	
027	022	RAL	
030	022	RAL	
031	022	RAL	
032	022	RAL	Check for bit 4
033	022	RAL	
034	100	JNC	
035	026		
036	000		
037	006	MVI A	
040	003	03	Display event 3
041	121	A → P0	
042	106	CALL	Delay 15 seconds
043	100	DELAY	
044	000		
045	006	MVI A	Raise discrete 6
046	040	040	
047	123	A → P01	
050	006	MVI A	
051	004	04	Display event 4
052	121	A → P0	
053	101	P0 → A	
054	022	RAL	
055	022	RAL	Check for bit 7
056	100	JNC	
057	053		
060	000		
061	006	MVI A	
062	005	05	Display event 5
063	121	A → P0	
064	104	JPU	
065	064		
066	000		

DELAY SUBROUTINE

LOCATION	OP CODE	OPERATION
100	330	A D
101	016	LDB I
102	377	COUNT
103	046	LDEI
104	000	ZERO
105	006	MVI A
106	000	
107	004	ADI
110	001	
111	140	JPC
112	117	117
113	000	000
114	104	JPU
115	107	105
116	000	000
117	304	E A
120	004	ADI
121	001	
122	140	JPC
123	131	131
124	000	000
125	340	A E
126	104	JPU
127	105	105
130	000	000
131	011	DCR B
132	150	JP0
133	140	140
134	000	000
135	104	JPU
136	105	105
137	000	000
140	303	D A
141	007	RET

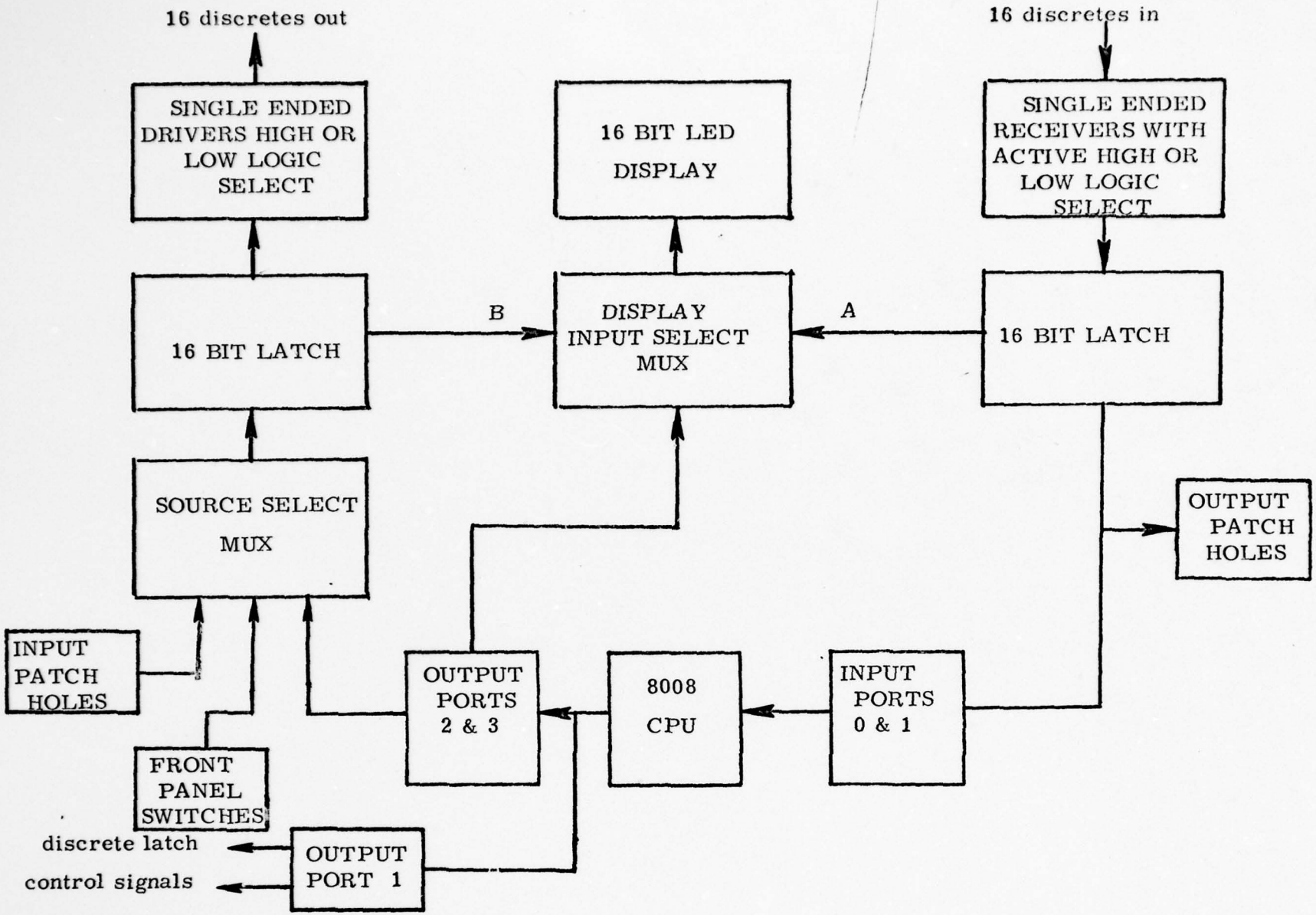


FIGURE D.1 CHECKOUT CHASSIS FUNCTIONAL BLOCK DIAGRAM
(DISCRETE HANDLING AND DISPLAY)

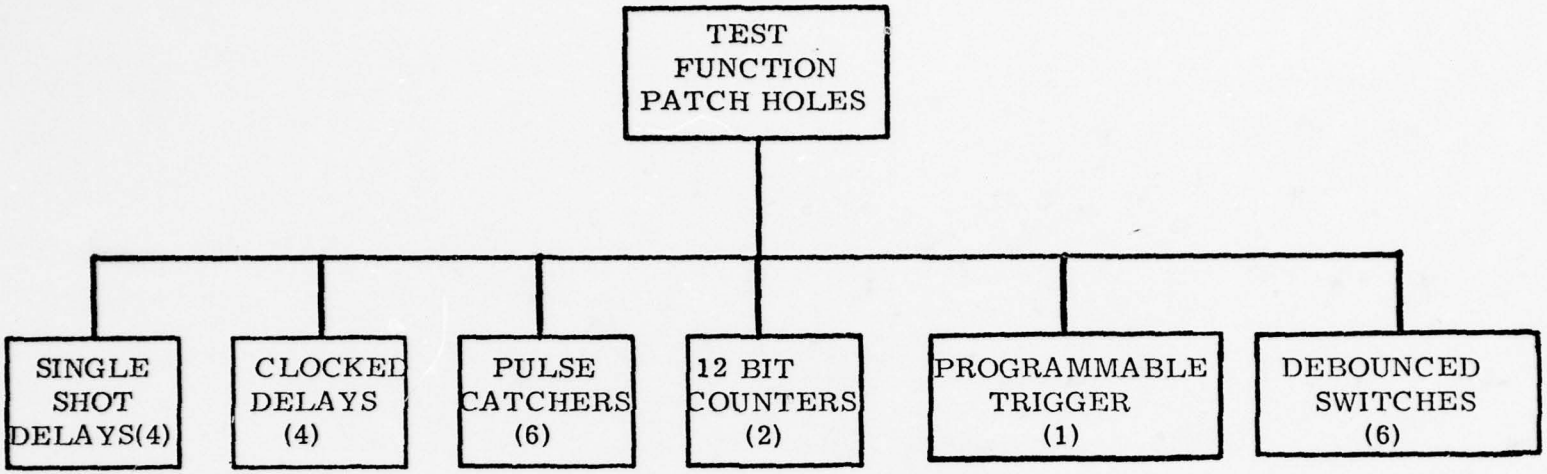


FIGURE D. 2

CHECKOUT CHASSIS FUNCTIONAL PLOCK DIAGRAM

(TEST FUNCTIONS)

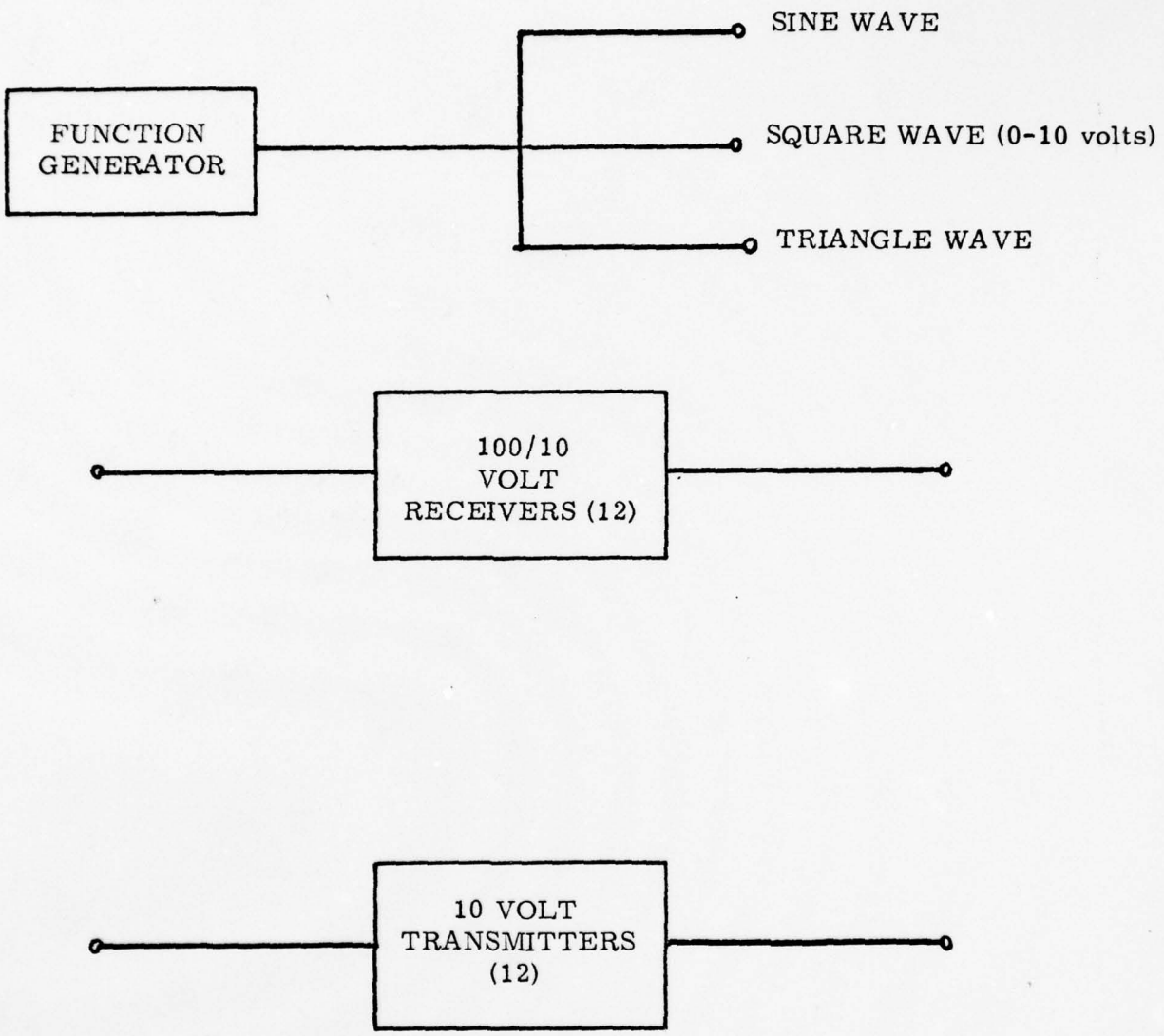
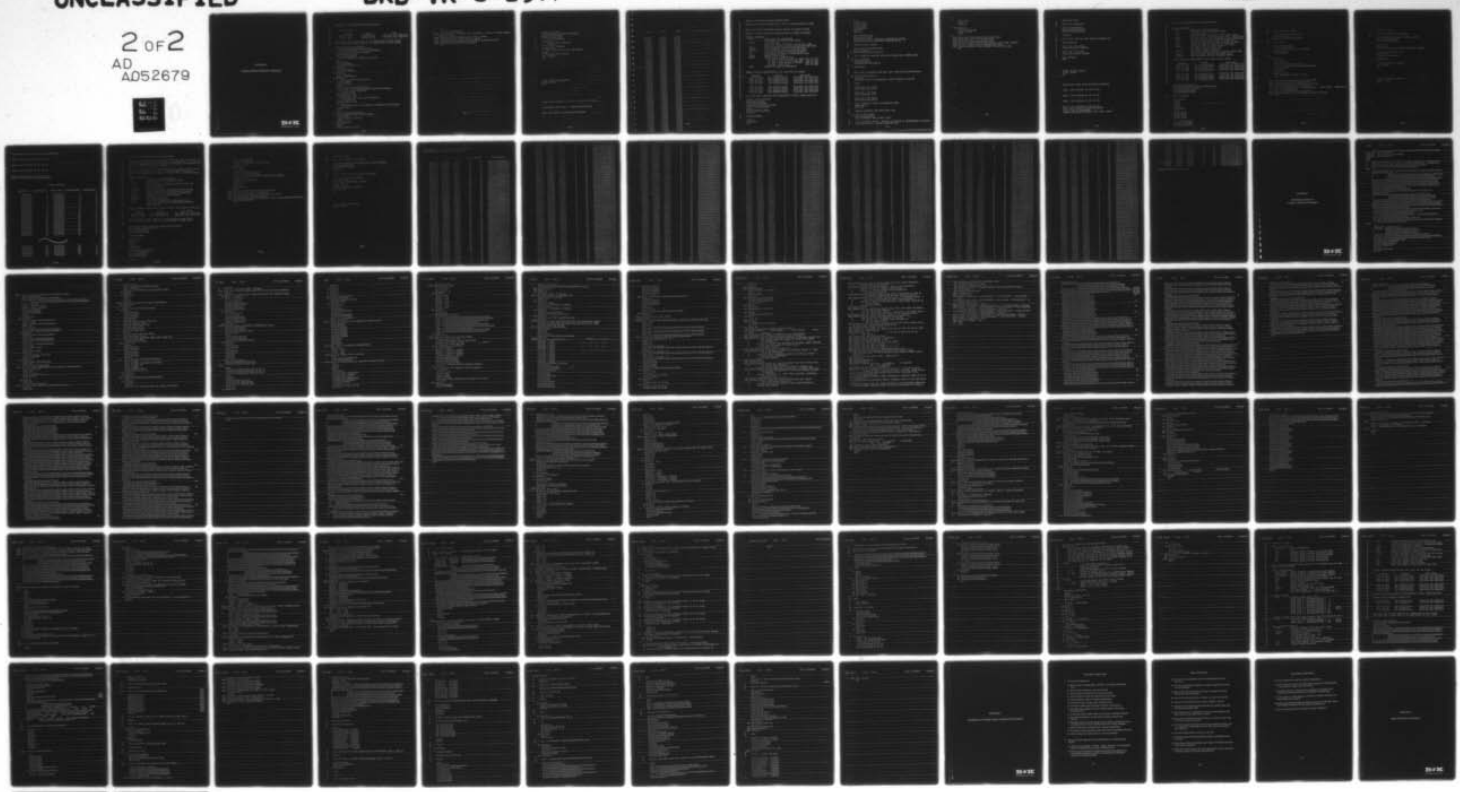


FIGURE D.3 CHECKOUT CHASSIS FUNCTIONAL BLOCK DIAGRAM (ANALOG FUNCTIONS)

AD-A052 679 B-K DYNAMICS INC ROCKVILLE MD F/G 9/2
QUARTERLY INTERIM TECHNICAL REPORT (3RD), CONTRACT DAAH01-75-C---ETC(U)
JUL 75 DAAH01-75-C-0194
UNCLASSIFIED BKD-TR-3-197. NL

2 OF 2
AD
AD52679



END
DATE
FILMED
5-78
DDC

APPENDIX E

MISCELLANEOUS CHECKOUT PROGRAMS

BOK
DYNAMICS, INC.

```
      PROGRAM TSTDUN(OUTPUT,4FILE,TAPE6=OUTPUT)
C
C
C         TRUNK SET-UP
C
C            TRUNKING              FORTRAN                    AD/4
C          W-07 TO W-18          /*DAC1/49,DAC          TR50-TR57 AND TR70-TR77
C          W-05 TO W-03          /*ADC1/49,ADC          TR10-TR17 AND TR30-TR37
C
C     THE FIRST ADC IN EACH GROUP OF 16 CORRESPONDS TO AD/4 TR-10.
C     THE FIRST DAC IN EACH GROUP OF 16 CORRESPONDS TO AD/4 TR-50.
C
      INTEGER PATIN(2),PATOUT(2)
      COMMON/DUTIN/DTDA(402),ATDD(402),I
C           SET UP INTERRUPT AND DADIOS VARIABLES
      INTERRUPT(I=1,R=20,T=100,P=1000)
      COMMON/*DAC1/49,DAC
      COMMON/*ADC1/49,ADC
C           SET UP ARRAY OF NUMBERS TO BE SENT
      I=1
      DTDA(1)=1.0
      DO 10 J=1,199,2
      DTDA(J+2)=DTDA(J)-.01
   10 CONTINUE
      DO 20 J=1,199,2
      DTDA(J+1)=-DTDA(J)
   20 CONTINUE
      DO 30 J=1,201
      DTDA(J+201)=DTDA(J)
   30 CONTINUE
C           READ AND PRINT TRUNK SET-UP
      PATIN(1)=3RV07
      PATIN(2)=3RV06
      LEN=2
      CALL PATSTAT(PATIN,PATOUT,LEN)
      WRITE(6,1000) PATIN(1),PATOUT(1),PATIN(2),PATOUT(2)
C           RESERVE HYBRID EQUIPMENT
      CALL RESERVE(IERR)
      WRITE(6,2000) IERR
      IF(IERR.NE.0) STOP
C           ENTER REAL TIME AND HOLD BACKGROUND
      CALL SIMRUN(ISTAT)
      WRITE(6,3000) ISTAT
      CALL BHOLD
C           PRINT TRANSMITTED AND RECEIVED NUMBERS AND DIFFERENCES
      WRITE(6,4000)
      LINE=1
      J=1
   40 DIFF=ABS(DTDA(J)-ATDD(J))
      WRITE(6,5000) DTDA(J),ATDD(J),DIFF
      LINE=LINE+1
      IF(LINE.LE.54) GO TO 50
      LINE=1
      WRITE(6,4000)
   50 CONTINUE
      J=J+1
      IF(J.LE.402) GO TO 40
```

```
      STOP
C          FORMAT STATEMENTS
 1000 FORMAT(1H1,/////,* TRUNK LINE PATCHING IS *,R3,* TO *,R3,* AND *,
     1R3,* TO *,R3)
 2000 FORMAT(///,* RESERVATION ERROR CODE = *,O20)
 3000 FORMAT(///,* REAL TIME STATUS = *,O20)
 4000 FORMAT(1H1,/////,5X,*O TO A*,7X,*A TO O*,9X,*DIFF*,//)
 5000 FORMAT(1X,F10.5,3X,F10.5,3X,F10.5)
      END
```

```
      SUBROUTINE CONVRT
      COMMON/OUTIN/DTDA(402),ATOD(402),I
      COMMON/*DAC1/48,DAC
      COMMON/*ADC1/48,ADC
C         D TO A CONVERSION
      DAC=DTDA(I)
C         WAIT FOR NEXT INTERRUPT
      CALL SIMWAIT
C         A TO D CONVERSION
      ATOD(I)=ADC
C         SET UP FOR NEXT D TO A CONVERSION
      I=I+1
      IF(I.GT.402) CALL SIMSTOP
      CALL SIMIDLE
      END
```

```
      TREE TSTCON(0),CONVRT(1)
      GLOBAL OUTIN
      END
```

```
      TRUNK LINE PATCHING IS V07 TO W13 AND V05 TO W03

      RESERVATION ERROR CODE = 00000000000000000000

      REAL TIME STATUS = 00000000000000000000
```

D TO A	A TO D	DIFF
1.00000	.99975	.00024
-1.00000	-.99994	.00005
.99000	.98969	.00031
-.99000	-.99994	.00994
.98000	.97974	.00026
-.98000	-.99994	.01994
.97000	.96973	.00027
-.97000	-.99994	.02994
.96000	.95965	.00034
-.96000	-.99994	.03994
.95000	.94958	.00042
-.95000	-.99994	.04994
.94000	.93964	.00035
-.94000	-.99994	.05994
.93000	.92969	.00031
-.93000	-.99994	.06994
.92000	.91974	.00026
-.92000	-.99994	.07994
.91000	.90967	.00033
-.91000	-.99994	.08994
.90000	.89972	.00028
-.90000	-.99994	.09994
.89000	.88971	.00029
-.89000	-.99994	.10994
.88000	.87970	.00030
-.88000	-.99994	.11994
.87000	.86975	.00025
-.87000	-.99994	.12994
.86000	.85968	.00032
-.86000	-.99994	.13994
.85000	.84973	.00027
-.85000	-.99994	.14994
.84000	.83972	.00028
-.84000	-.99994	.15994
.83000	.82965	.00035
-.83000	-.99994	.16994
.82000	.81964	.00036
-.82000	-.99994	.17994
.81000	.80975	.00025
-.81000	-.99994	.18994
.80000	.79974	.00026
-.80000	-.99994	.19994
.79000	.78967	.00033
-.79000	-.99994	.20994
.78000	.77960	.00040
-.78000	-.99994	.21994
.77000	.76971	.00029
-.77000	-.99994	.22994
.76000	.75970	.00030
-.76000	-.99994	.23994
.75000	.74982	.00018
-.75000	-.99994	.24994

```
      PROGRAM TRJAM(OUTPUT,HFILE,TAPE6=OUTPUT)
C
C     USE ONLY ONE INPUT DISCRETE AT A TIME TO AVOID ERRORS IN LOGIC
C
C
C     NOTE, ALL OUTPUT DISCRETES SHOULD APPEAR ON DISPLAY LIGHTS
C
C     PROGRAM FOR NON REAL TIME COMMUNICATION OF DADIOS VARIABLES
C
C     PROGRAM VARIABLES
C        IERR          ERROR CODE FOR RESERVATION
C                      0=NOERROR, GT.0=RESERVATION ERROR
C        ISTAT         REAL TIME MODE
C                      0=IN REAL TIME, ISTAT.GT.0 NOT IN REAL TIME
C        LEN           NUMBER OF TRUNKING STATION CONNECTOR PAIRS USED
C        PIN(I)        I-TH TRUNCKING STATION CONNECTOR INPUT
C        POUT(I)       I-TH TRUNCKING STATION CONNECTOR OUTPUT
C        LARA          AN ARRAY OF LOGICAL INTERRUPT NUMBERS
C        N             THE NUMBER OF LOGICAL INTERRUPTS IN LARA
C        IODIS         DADIOS OUTPUT WORD
C        IIDIS         DADIOS INPUT DISCRETE WORD
C                         IF LINE 1 HIGH VARY IOUT IN BATCH  TR27 OR TR67
C                         IF LINE 2 HIGH GO INTO REAL TIME   TR26 OR TR66
C                         IF LINE 3 HIGH RETURN TO BKGRND    TR25 OR TR65
C                         IF LINE 4 HIGH STOP                TR24 OR TR64
C        LOOP          NUMBER OF LOOPS IN BACKGROUND
C
C
C     DADIOS PATCHING REQUIREMENTS (ONE IIDIS AND ONE IODIS)
C
C        TRUNKING              FORTRAN                AD/4 PATCH
C     V-50 TO W-50      FOR /IOIS2/1,IIDIS     TR00-TR07 AND TR20-TR27
C     V-52 TO W-50      FOR /IOIS2/1,IIDIS     TR40-TR47 AND TR60-TR67
C     V-50 TO W-51      FOR /IOIS2/2,IID S     TR00-TR07 AND TR20-TR27
C     V-52 TO W-51      FOR /IOIS2/2,IIDIS     TR40-TR47 AND TR60-TR67
C
C     V-51 TO W-60      FOR /OOIS2/1,IODIS     TR10-TR17 AND TR30-TR37
C     V-53 TO W-60      FOR /OOIS2/1,IODIS     TR50-TR57 AND TR70-TR77
C     V-51 TO W-61      FOR /OOIS2/2,IODIS     TR10-TR17 AND TR30-TR37
C     V-53 TO W-61      FOR /OOIS2/2,IODIS     TR50-TR57 AND TR70-TR77
C
C     THE HIGH ORDER CDC-6600 BIT CORRESPONDS TO TR0X, WHERE X=0,2,4,6
C
      INTEGER PIN(8),POUT(8)
      DIMENSION LARA(8)
      COMMON/INTCOM/ICNT1
      INTERRUPT(I=1,R=20,T=100000)
      COMMON/*IOIS2/1,IIDIS
      JAM,ON
      COMMON/*OOIS2/1,IODIS
      JAM,OFF
C
C     INITIALIZATION
C
      LOOP=0
      LARA(1)=1
      N=1
```

```
      LEN=4
      PIN(1)=3RV50
      PIN(2)=3RV51
      PIN(3)=3RV52
      PIN(4)=3RV53
      ICCUNT=0
      ICNT1=0
C
      CALL RESERVE(IERR)
      WRITE(6,1000)IERR
      IF(IERR.NE.0)CALL REMARK(18H RESERVATION ERROR)
      IF(IERR.EC.0)CALL REMARK(18H RESERVATION OK    )
C
C     CHECK PATCHING STATUS
C
      CALL PATSTAT(PIN,POUT,LEN)
      DO 10 I=1,LEN
   10 WRITE(6,8000)PIN(I),POUT(I)
C
C     JAM A DISCRETE WORD OUT ONTO AD-4 TRUNK LINE IN BATCH MODE
C
   20 IOUT=2**ICOUNT
      ICCUNT=ICCUNT+1
      IF(ICOUNT.EQ.16)ICOUNT=0
C
      IODIS=IOUT
C
C
C     READ BACK A DISCRETE WORD FROM AD-4 TRUNK LINE IN BATCH MODE
C
   30 CALL INTCNVT(LARA,N)
      IN=IIDIS
C     PAUSE AND WAIT FOR APPROPRIATE INPUT DISCRETE TO BE SET
C
      PAUSE
C
C     TEST LINE 1 OF IIDIS
      IF(IN.EQ.1B)GO TO 20
C
C     TEST LINE 2 OF IIDIS
      IF(IN.EQ.2B)GO TO 40
C
C     TEST LINE 4 OF IIDIS
      IF(IN.EQ.10B)GO TO 50
C
C     COUNT NUMBER OF LOOPS IN BACKGROUND MODE
      LOOP=LOOP+1
      GO TO 30
C
C     RESERVE EQUIPMENT AND ENTER REAL TIME
C
   40 CALL SIMRLN(ISTAT)
      WRITE(6,2000)ISTAT
      CALL REMARK(17H JOB IN REAL TIME)
C
C     HOLD BACKGROUND UNTILL  SIMHOLC OR SIMSTOP IS INCOUNTERED, ON RETURN
C     TO BATCH TURN ALL  OUTPUT DISCRETES OFF
```

```
C
      CALL BHOLD
      IODIS=0
      GO TO 30
C
   50 CONTINUE
      WRITE(6,5000)LOOP
      WRITE(6,4000)
      STOP
C
 1000 FORMAT(24H RESERVATION ERROR CODE=,O20)
 2000 FORMAT(18H REAL TIME STATUS=,O20)
 3000 FORMAT(4H IN,,O20,2X,5HICUT=,O20)
 4000 FORMAT(* PROGRAM TERMINATED NORMALLY WITH LINE 4 HIGH*)
 5000 FORM T(* TCTAL LCOPS IN BACKGROUND=*,I20)
 8000 FORMAT(/*0TRUNK  LINE PATCHING IS *,R3,* TO *,R3)
      END
```

```
      SUBROUTINE SUB1
C
C     REAL TIME SUBROUTINE
C
      COMMON/INTCOM/ICNT1
      COMMON/*ICIS2/1,IIDIS
      COMMON/*ODIS2/1,IODIS
C
      IN=IIDIS
C
C     IF IN REAL TIME TURN ALL OUTPUT DISCRETES ON
C
      IODIS=2**16-1
C
C     TEST LINE 3 OF IIDIS
      IF(IN.EQ.4B)CALL SIMSTOP
C
C     TEST LINE 4 OF IIDIS
      IF(IN.EQ.10B)CALL SIMSTOP
C
      CALL SIMIOLE
      END



      RTREE TRJAM(0),SUB1(1)
      GLOBAL INTCOM
      END



      RESERVATION ERROR CODE=0000000000000000000

      TRUNK LINE PATCHING IS V50 TO W50

      TRUNK LINE PATCHING IS V51 TO W60

      TRUNK LINE PATCHING IS V52 TO W51

      TRUNK LINE PATCHING IS V53 TO W)F
      REAL TIME STATUS=0000000000000000000
      TOTAL LOOPS IN BACKGROUND=                    3
      PROGRAM TERMINATED NORMALLY WITH LINE 4 HIGH
```

```
      PROGRAM TRDISID(OUTPUT,TAPE6=OUTPUT)


C     PROGRAM VARIABLES
C        IERR          ERROR CODE FOR RESERVATION
C                      0=NOERROR, GT.0=RESERVATION ERROR
C        ISTAT         REAL TIME MODE
C                      0=IN REAL TIME, ISTAT.GT.0 NOT IN REAL TIME
C        IIDIS         CDC-6600 SENSE LINE DISCRETE(16 BIT)  IIDIS=IBACK
C        IODIS         CDC-6600 CONTROL LINE DISCRETE(16 BIT) IODIS=IOUT
C        MAX           DEC. EQUIVALENT OF 16 BITS ALL EQUAL ONE
C        LOOP          NUMBER OF INTERRUPTS BEFORE EQUALITY OF BITS
C        LINE          NUMBER OF LINES OF PRINTOUT IN EXECUTION
C        IWRITE        =0, WRITE ONLY IF ERRORS OCCUR
C                      =1, WRITE ON EVERY ITERATION
C        MAX_LINE      MAX NUMBER LINES OF OUTPUT
C        LEN           NUMBER OF TRUNKING STATION CONNECTOR PAIRS USED
C        PIN(I)        I-TH TRUNKING STATION CONNECTOR INPUT
C        POUT(I)       I-TH TRUNKING STATION CONNECTOR OUTPUT

C     DADIOS PATCHING REQUIREMENTS (ONE IIDIS AND ONE IODIS)

C        TRUNKING              FORTRAN                 AD/4 PATCH
C     W-50 TO W-50      FOR /IDIS2/1,IIDIS      TR00-TR07 AND TR20-TR27
C     W-52 TO W-50      FOR /IDIS2/1,IIDIS      TR40-TR47 AND TR60-TR67
C     W-50 TO W-51      FOR /IDIS2/2,IIDIS      TR00-TR07 AND TR20-TR27
C     W-52 TO W-51      FOR /IDIS2/2,IIDIS      TR40-TR47 AND TR60-TR67
C
C     W-51 TO W-50      FOR /ODIS2/1,IODIS      TR10-TR17 AND TR30-TR37
C     W-53 TO W-50      FOR /ODIS2/1,IODIS      TR50-TR57 AND TR70-TR77
C     W-51 TO W-51      FOR /ODIS2/2,IODIS      TR10-TR17 AND TR30-TR37
C     W-53 TO W-51      FOR /ODIS2/2,IODIS      TR50-TR57 AND TR70-TR77


      INTEGER PIN(8),POUT(8)
      COMMON/INTCOM/IOUT,LOOP,MAX,IBACK,IWRITE
      INTERRUPT(I=1,R=10,T=500)
      COMMON/*IDIS2/1,IIDIS
      COMMON/*ODIS2/1,IODIS

C     INITIALIZATION

      MAX=2**16-1
      IOUT=0
      LOOP=0
      LINE=0
      IWRITE=1
      MAX_LINE=400
      LEN=4
      PIN(1)=3R/50
      PIN(2)=3R/51
      PIN(3)=3R/52
      PIN(4)=3R/53

      CALL RESERVE(IERR)
      WRITE(6,1000)IERR
      IF(IERR.NE.0)STOP
```

```
C
C     CHECK PATCHING STATUS
C
      CALL PATSTAT(PIN,POUT,LEN)
      DO 10 I=1,LEN
   10 WRITE(6,3000)PIN(I),POUT(I)
C
C     REAL TIME
C
      CALL SIMRUN(ISTAT)
      WRITE(6,2000)ISTAT
      CALL REMARK(17H JOB IN REAL TIME)
      WRITE(6,3000)
      WRITE(6,3000)
C
C
   25 CONTINUE
      CALL SHOLD
      LINE=LINE+1
      IF(LINE.GT.MAXLINE)GO TO 20
      WRITE(6,4000)IOUT,IOUT,IBACK,IBACK,LOOP
      CALL SIM MOVE
      CALL SIMGO
      GO TO 25
   20 WRITE(6,3000)
      CALL REMARK(18H RETURN TO MAIN)
      STOP
C
 1000 FORMAT(24H1RESERVATION ERROR CODE=,I20)
 2000 FORMAT(18H REAL TIME STATUS=,I20)
 3000 FORMAT(1H0,*PROGRAM TERMINATED NORMALLY*)
 4000 FORMAT(10X,O10,I15,5X,O10,I15,I15)
 5000 FORMAT(5X,* IOUT(OCTAL)   IOUT(DECIMAL)   IBACK(OCTAL)  IBACK(DECI
     1MAL)  LOOP(DECIMAL)*/)
 6000 FORMAT(///,35X,*ERRORS DETECTED*//)
 8000 FORMAT(/*0TRUNK  LINE PATCHING IS *,R3,* TO *,R3)
      END
```

```
      SUBROUTINE SUB1
C
C     REAL TIME INTERRUPT SUBROUTINE
C
      COMMON/INTCOM/IOUT,LOOP,MAX,IBACK,IWRITE
      COMMON/*IDIS2/1,IIDIS
      COMMON/*ODIS2/1,IODIS
C
      IODIS=IOUT
      IBACK=IIDIS
      IF(IOUT.EQ.IBACK.AND.IWRITE.EQ.1)CALL SIMHLD
C
      IF(IOUT.NE.IBACK)GO TO 10
      IOUT=IOUT+1
      IF(IOUT.EQ.MAX)IOUT=1
      LOOP=1
   10 LOOP=LOOP+1
C
      IF(LOOP.EQ.10)CALL SIMHLD
      CALL SIMIDLE
      END



      RTREE TRJISR(0),SUB1(1)
      GLOBAL INTCOM
      END
```

```
RESERVATION ERROR CODE=0000000000000000000

TRUNK LINE PATCHING IS V60 TO N60

TRUNK LINE PATCHING IS V61 TO N60

TRUNK LINE PATCHING IS V62 TO N61

TRUNK LINE PATCHING IS V63 TO N)F
REAL TIME STATUS=0000000000000000002
```

ERRORS DETECTED

IOUT(OCTAL)	IOUT(DECIMAL)	IBACK(OCTAL)	IBACK(DECIMAL)	LOOP(DECIMAL)
0000000000	0	0000000000	0	0
0000000001	1	0000000001	1	3
0000000002	2	0000000002	2	3
0000000003	3	0000000003	3	3
0000000004	4	0000000004	4	3
0000000005	5	0000000005	5	3
0000000006	6	0000000006	6	3
0000000007	7	0000000007	7	3
0000000010	8	0000000010	8	3
0000000011	9	0000000011	9	3
0000000012	10	0000000012	10	3
0000000013	11	0000000013	11	3
0000000014	12	0000000014	12	3
0000000015	13	0000000015	13	3
0000000016	14	0000000016	14	3
0000000017	15	0000000017	15	3
0000000020	16	0000000020	16	3
0000000021	17	0000000021	17	3
0000000022	18	0000000022	18	3
0000000023	19	0000000023	19	3
0000000024	20	0000000024	20	3
0000000025	21	0000000025	21	3
0000000026	22	0000000026	22	3

0000000373	251	0000000373	251	3
0000000374	252	0000000374	252	3
0000000375	253	0000000375	253	3
0000000376	254	0000000376	254	3
0000000377	255	0000000377	255	3
0000000400	256	0000000000	0	10

```
      PROGRAM TRALBIO(OUTPUT,HFILE,TAPE6=OUTPUT)
C
C     PROGRAM TO TEST ANALOG SIGNALS BETWEEN AD/4 AND CDC/6600. THIS TASK
C     IS ACCOMPLISHED BY TURNING THE ANALOG SIGNAL AROUND AT THE AD/4 AND
C     COMPARING DIFFERENCE UPON RETURN TO THE CDC/6600. THE PROGRAM TEST
C     FOR ERRORS GREATER THAN FIVE PERCENT.
C
C     NOTE ADCS AND DACS ARE IN GROUPS OF 16, 1-15, 17-32, 33-48, 49-64.
C     FLOATING POINT ANALOG SIGNALS ARE SCALED GE -1.0 AND LE +1.0.
C     INTEGER ANALOG SIGNALS ARE SCALED GE -32767 AND LE +32767(14 BIT).
C
C     PROGRAM VARIABLES
C        IERR           ERROR CODE FOR RESERVATION
C                       0=NOERROR, CT.0=RESERVATION ERROR
C        ISTAT          REAL TIME MODE
C                       0=IN REAL TIME, ISTAT.GT.0 NOT IN REAL TIME
C        OUT            THE DAC VARIABLE
C        BACK           THE ADC VARIABLE
C        BIGDIFF        MAXIMUM ALLOWABLE DIFFERENCE IN ADC AND DAC
C        DIFF           ACTUAL COMPUTED DIFFERENCE IN ADC AND DAC
C        LINE           NUMBER OF LINES OF PRINTOUT IN EXECUTION
C        MAXLINE        MAXIMUM NUMBER OF LINES PRINTED
C        TIMELFT        REMAINING RCT TIME
C        STATUS         THE HRTM STATUS WORD
C        LOOP           NUMBER OF ATTEMPTS TO EQUATE ADC AND DAC
C        J              A MASKING  WORD USED IN CONJUNCTION WITH THE
C                       STAT FUNCTION
C
C     DADIOS PATCHING REQUIREMENTS (AD/4 FIELD 3, 4TH GROUP ADC AND DAC)
C
C        TRUNKING              FORTRAN                AD/4 PATCH
C        W-08 TO V-06     FOR /*ADC1/49,ADC      TR10-TR17 AND TR30-TR37
C        W-13 TO V-07     FOR /*DAC1/49,DAC      TR50-TR57 AND TR70-TR77
C
C     THE FIRST ADC IN EACH GROUP OF 16 CORRESPONDS TO AD/4 TR-10.
C     THE FIRST DAC IN EACH GROUP OF 16 CORRESPONDS TO AD/4 TR-50.
C
C
      COMMON/INTCOM/OUT,LOOP,BACK,BIGDIFF,DIFF,STATUS,J
      INTERRUPT(I=1,R=100,T=10000)
      COMMON/*ADC1/49,ADC
      COMMON/*DAC1/49,DAC
C
C     INITIALIZATION
C
      BIGDIFF=.01
      OUT=-1.5
      LOOP=0
      LINE=0
      MAXLINE=400
      IWRITE=0
      J=777777777777777777777B
      CALL RESERVE(IERR)
      WRITE(6,1000)IERR
      IF(IERR.NE.0)STOP
C
C     REAL TIME
```

```
C
      CALL SIMRUN(ISTAT)
      WRITE(6,2000)ISTAT
      CALL REMARK(17H JOB IN REAL TIME)
      WRITE(6,5000)
   25 CONTINUE
      CALL SHOLD
      LINE=LINE+1
      IF(LINE.GT.MAXLINE)GO TO 50
      TIMELFT=STATUS.A.37777770
      WRITE(6,4000)OUT,BACK,DIFF,LOOP,TIMELFT,STATUS
      OUT=OUT+0.01
      IF(OUT.GT.1.50)OUT=0.0
      LOOP=0
      CALL SIMGO
      GO TO 25
   50 WRITE(6,3000)
      STOP
 1000 FORMAT(24H1RESERVATION ERROR CODE=,O20)
 2000 FORMAT(15H REAL TIME STATUS=,O20)
 3000 FORMAT(1H0,*PROGRAM TERMINATED ON MAX LINE*)
 4000 FORMAT(5X,3F10.6,I10,I10,5X,O20)
 5000 FORMAT(///12X,*OUT*,8X,*BACK*,4X,* DIFF *,5X,*LOOP*,3X,*TIMELFT*,
     1 8X,*STATUS(OCTAL)*/)
      END
```

```
      SUBROUTINE SUB1
C
C     REAL TIME INTERRUPT SUBROUTINE
C
      COMMON/INTCOM/OUT,LOOP,BACK,BIGDIFF,DIFF,STATUS,J
      COMMON/*ADC1/49,ADC
      COMMON/*DAC1/49,DAC
      DAC=OUT
      BACK=ADC
C
C     COMPUTE DIFFERENCE IN ADC AND DAC VALUES
C
      DIFF=ABS(ABS(OUT)-ABS(BACK))
      STATUS=STAT(J)
      IF(DIFF.LT.BIGDIFF)CALL SIMHOLD
      LOOP=LOOP+1
      STATUS=STAT(J)
      IF(LOOP.EQ.10)CALL SIMHOLD
      CALL SIMIBLE
      END



      RTREF IRALOID(0),SUB1(1)
      GLOBAL INTCOM
      END
```

RESERVATION ERROR CODE=000000000000000000000
REAL TIME STATUS=000000000000000000000

OUT	BACK	DIFF	LOOP	TIME_ET	STATUS(OCTAL)
-1.500000	-.999994	.500006	10	98	3074000000000000000142
-1.490000	-.999994	.490006	10	98	3074000000000000000142
-1.480000	-.999994	.480006	10	98	3074000000000000000142
-1.470000	-.999994	.470006	10	98	3074000000000000000142
-1.460000	-.999994	.460006	10	98	3074000000000000000142
-1.450000	-.999994	.450006	10	98	3074000000000000000142
-1.440000	-.999994	.440006	10	98	3074000000000000000142
-1.430000	-.999994	.430006	10	98	3074000000000000000142
-1.420000	-.999994	.420006	10	98	3074000000000000000142
-1.410000	-.999994	.410006	10	99	3074000000000000000143
-1.400000	-.999994	.400006	10	98	3074000000000000000142
-1.390000	-.999994	.390006	10	98	3074000000000000000142
-1.380000	-.999994	.380006	10	99	3074000000000000000143
-1.370000	-.999994	.370006	10	98	3074000000000000000142
-1.360000	-.999994	.360006	10	98	3074000000000000000142
-1.350000	-.999994	.350006	10	98	3074000000000000000142
-1.340000	-.999994	.340006	10	98	3074000000000000000142
-1.330000	-.999994	.330006	10	98	3074000000000000000142
-1.320000	-.999994	.320006	10	98	3074000000000000000142
-1.310000	-.999994	.310006	10	98	3074000000000000000142
-1.300000	-.999994	.300006	10	98	3074000000000000000142
-1.290000	-.999994	.290006	10	98	3074000000000000000142
-1.280000	-.999994	.280006	10	98	3074000000000000000142
-1.270000	-.999994	.270006	10	98	3074000000000000000142
-1.260000	-.999994	.260006	10	98	3074000000000000000142
-1.250000	-.999994	.250006	10	98	3074000000000000000142
-1.240000	-.999994	.240006	10	98	3074000000000000000142
-1.230000	-.999994	.230006	10	98	3074000000000000000142
-1.220000	-.999994	.220006	10	98	3074000000000000000142
-1.210000	-.999994	.210006	10	98	3074000000000000000142
-1.200000	-.999994	.200006	10	98	3074000000000000000142
-1.190000	-.999994	.190006	10	98	3074000000000000000142
-1.180000	-.999994	.180006	10	98	3074000000000000000142
-1.170000	-.999994	.170006	10	98	3074000000000000000142
-1.160000	-.999994	.160006	10	98	3074000000000000000142
-1.150000	-.999994	.150006	10	98	3074000000000000000142
-1.140000	-.999994	.140006	10	98	3074000000000000000142
-1.130000	-.999994	.130006	10	98	3074000000000000000142
-1.120000	-.999994	.120006	10	98	3074000000000000000142
-1.110000	-.999994	.110006	10	98	3074000000000000000142
-1.100000	-.999994	.100006	10	98	3074000000000000000142
-1.090000	-.999994	.090006	10	98	3074000000000000000142
-1.080000	-.999994	.080006	10	98	3074000000000000000142
-1.070000	-.999994	.070006	10	98	3074000000000000000142
-1.060000	-.999994	.060006	10	98	3074000000000000000142
-1.050000	-.999994	.050006	10	98	3074000000000000000142
-1.040000	-.999994	.040006	10	98	3074000000000000000142
-1.030000	-.999994	.030006	10	98	3074000000000000000142
-1.020000	-.999994	.020006	10	98	3074000000000000000142
-1.010000	-.999994	.010006	10	98	3074000000000000000142
-1.000000	-.999994	.000006	0	99	1074000000000000000143
-.999100	-.999094	.000994	1	99	1074000000000000000143
-.988100	-.988110	.000010	3	99	0074000000000000000143
-.977000	-.977003	.000003	3	99	0074000000000000000143
-.965000	-.965007	.000007	2	99	0074000000000000000143
-.955000	-.955007	.000007	3	99	0074000000000000000143

-.93000	-.[illegible]	.[illegible]	[illegible]	[illegible]	[illegible]
-.92000	-.92004	.00004	3	99	0074000100000000143
-.91000	-.91998	.01118	1	99	0074000000000000143
-.90000	-.89997	.01997	1	99	0074000000000000143
-.89000	-.88995	.10005	3	99	0074001000000000143
-.88000	-.88995	.00105	1	99	0074000000000000143
-.87000	-.87994	.00994	1	99	0074000000000000143
-.86000	-.86006	.00006	3	99	0074000000000000143
-.85000	-.85999	.00001	1	99	0074000000000000143
-.84000	-.84998	.00998	1	99	0074000000000000143
-.83000	-.83997	.00997	1	99	0074000000000000143
-.82000	-.82001	.00001	3	99	0074000000000000143
-.81000	-.81995	.00995	2	99	0074000000000000143
-.80000	-.81000	.01000	1	99	0074000000000000143
-.79000	-.79999	.00999	2	99	0074000000000000143
-.78000	-.78998	.00998	2	99	0074000000000000143
-.77000	-.77997	.00997	1	99	0074000000000000143
-.76000	-.75996	.01996	2	99	0074000000000000143
-.75000	-.75989	.00389	1	99	0074000000000000143
-.74000	-.74005	.00005	3	99	0074000000000000143
-.73000	-.73010	.00010	3	99	0074000000000000143
-.72000	-.72009	.00009	3	99	0074000000000000143
-.71000	-.71997	.00997	2	99	0074000000000000143
-.70000	-.70001	.00001	3	99	0074000000000000143
-.69000	-.68995	.00995	2	99	0074000000000000143
-.68000	-.68018	.00018	3	99	0074000000000000143
-.67000	-.67004	.00004	3	99	0074000000000000143
-.66000	-.66010	.00010	3	99	0074000000000000143
-.65000	-.64990	.00010	3	99	0074000000000000143
-.64000	-.64001	.00001	3	99	0074000000000000143
-.63000	-.63995	.00995	1	99	0074000000000000143
-.62000	-.61988	.00988	2	99	0074000000000000143
-.61000	-.61005	.00005	3	99	0074000000000000143
-.60000	-.60004	.00004	3	99	0074000000000000143
-.59000	-.58997	.00003	3	99	0074000000000000143
-.58000	-.57996	.00004	3	99	0074000000000000143
-.57000	-.57996	.00996	1	99	0074000000000000143
-.56000	-.56000	.00000	3	99	0074000000000000143
-.55000	-.56000	.01000	1	99	0074000000000000143
-.54000	-.54999	.01999	2	99	0074000000000000143
-.53000	-.52997	.00003	3	99	0074000000000000143
-.52000	-.52997	.00997	2	99	0074000000000000143
-.51000	-.50995	.00005	3	99	0074000000000000143
-.50000	-.50995	.00995	1	99	0074000000000000143
-.49000	-.49017	.00017	3	99	0074000000000000143
-.48000	-.48015	.00015	3	99	0074000000000000143
-.47000	-.47009	.00009	3	99	0074000000000000143
-.46000	-.46002	.00002	3	99	0074000000000000143
-.45000	-.44996	.00996	2	99	0074000000000000143
-.44000	-.44000	.00000	3	99	0074000000000000143
-.43000	-.43011	.00011	3	99	0074000000000000143
-.42000	-.42017	.00017	3	99	0074000000000000143
-.41000	-.41016	.00016	3	99	0074000000000000143
-.40000	-.40002	.00002	3	99	0074000000000000143
-.39000	-.39001	.00001	3	99	0074000000000000143
-.38000	-.38007	.00007	3	99	0074000000000000143
-.37000	-.37018	.00018	3	99	0074000000000000143
-.36000	-.35995	.00005	3	99	0074000000000000143
-.35000	-.35015	.00015	3	99	0074000000000000143
-.34000	-.34015	.00015	3	99	0074000000000000143
-.33000	-.33008	.00008	3	99	0074000000000000143
-.32000	-.32001	.00001	3	99	0074000000000000143
-.31000	-.31005	.00005	3	99	0074000000000000143
-.30000	-.30011	.00011	3	99	0074000000000000143
-.29000	-.29004	.00004	3	99	0074000000000000143
-.28000	-.28003	.00003	3	99	0074000000000000143

-.27100	-.27093	.00007	3	99	0074000000000000143
-.26000	-.26001	.00001	3	99	0074000000000000143
-.25000	-.25000	.00000	3	99	0074000000000000143
-.24000	-.24011	.00011	3	99	0074000000000000143
-.23000	-.23010	.00010	3	99	0074000000000000143
-.22000	-.22003	.00003	3	99	0074000000000000143
-.21000	-.21002	.00002	3	99	0074000000000000143
-.20000	-.20013	.00013	3	99	0074000000000000143
-.19000	-.19006	.00006	3	99	0074000000000000143
-.18000	-.18018	.00018	3	99	0074000000000000143
-.17000	-.17010	.00010	3	99	0074000000000000143
-.16000	-.16010	.00010	3	99	0074000000000000143
-.15000	-.15002	.00002	3	99	0074000000000000143
-.14000	-.14001	.00001	3	99	0074000000000000143
-.13000	-.13007	.00007	3	99	0074000000000000143
-.12000	-.12006	.00006	3	99	0074000000000000143
-.11000	-.11011	.00011	3	99	0074000000000000143
-.10000	-.10010	.00010	3	99	0074000000000000143
-.09000	-.09003	.00003	3	99	0074000000000000143
-.08000	-.08014	.00014	3	99	0074000000000000143
-.07000	-.07001	.00001	3	99	0074000000000000143
-.06000	-.06012	.00012	3	99	0074000000000000143
-.05000	-.05000	.01000	1	99	0074000000000000143
-.04000	-.04010	.00010	3	99	0074000000000000143
-.03000	-.03003	.00003	3	99	0074000000000000143
-.02000	-.02008	.00008	3	99	0074000000000000143
-.01000	-.01001	.00001	3	99	0074000000000000143
.00000	-0.00000	.00000	3	99	0074000000000000143
.01000	.00983	.00017	3	99	0074000000000000143
.02000	.01984	.00016	3	99	0074000000000000143
.03000	.02979	.00021	3	99	0074000000000000143
.04000	.03992	.00008	3	99	0074000000000000143
.05000	.04993	.00007	3	99	0074000000000000143
.06000	.05988	.00012	3	99	0074000000000000143
.07000	.06982	.00018	3	99	0074000000000000143
.08000	.07990	.00010	3	99	0074000000000000143
.09000	.08984	.00016	3	99	0074000000000000143
.10000	.09985	.00015	3	99	0074000000000000143
.11000	.10980	.00020	3	99	0074000000000000143
.12000	.11993	.00007	3	99	0074000000000000143
.13000	.12982	.00018	3	99	0074000000000000143
.14000	.13983	.00017	3	99	0074000000000000143
.15000	.14984	.00016	3	99	0074000000000000143
.16000	.15991	.00009	3	99	0074000000000000143
.17000	.16992	.00008	3	99	0074000000000000143
.18000	.17999	.00001	3	99	0074000000000000143
.19000	.18994	.00006	3	99	0074000000000000143
.20000	.19995	.00005	3	99	0074000000000000143
.21000	.20990	.00010	3	99	0074000000000000143
.22000	.21979	.00021	3	99	0074000000000000143
.23000	.22992	.00008	3	99	0074000000000000143
.24000	.23987	.00013	3	99	0074000000000000143
.25000	.24994	.00006	3	99	0074000000000000143
.26000	.25983	.00017	3	99	0074000000000000143
.27000	.26984	.00016	3	99	0074000000000000143
.28000	.27985	.00015	3	99	0074000000000000143
.29000	.28986	.00014	3	99	0074000000000000143
.30000	.29993	.00007	3	99	0074000000000000143
.31000	.30988	.00012	3	99	0074000000000000143
.32000	.31982	.00018	3	99	0074000000000000143
.33000	.32983	.00017	3	99	0074000000000000143
.34000	.33984	.00016	3	99	0074000000000000143
.35000	.34991	.00009	3	99	0074000000000000143
.36000	.35999	.00001	3	100	0074000000000000144
.37000	.37000	.00000	3	99	0074000000000000143

.39000	.38993	.00007	3	99	0074000000000000000143
.40000	.39993	.00018	3	99	0074000000000000000143
.41000	.40993	.00015	3	99	0074000000000000000143
.42000	.41992	.00014	3	99	0074000000000000000143
.43000	.42993	.00007	3	99	0074000000000000000143
.44000	.43994	.00006	3	99	0074000000000000000143
.45000	.44989	.00011	3	99	0074000000000000000143
.46000	.45991	.00010	3	99	0074000000000000000143
.47000	.46987	.00013	3	99	0074000000000000000143
.48000	.47986	.00014	3	99	0074000000000000000143
.49000	.48993	.00007	3	99	0074000000000000000143
.50000	.49994	.00006	3	99	0074000000000000000143
.51000	.50983	.00017	3	99	0074000000000000000143
.52000	.51971	.00029	3	99	0074000000000000000143
.53000	.52979	.00021	3	99	0074000000000000000143
.54000	.53992	.00008	3	99	0074000000000000000143
.55000	.54993	.00007	3	99	0074000000000000000143
.56000	.55988	.00012	3	99	0074000000000000000143
.57000	.56970	.00030	3	99	0074000000000000000143
.58000	.57983	.00017	3	99	0074000000000000000143
.59000	.58984	.00016	3	99	0074000000000000000143
.60000	.59979	.00021	3	99	0074000000000000000143
.61000	.60980	.00020	3	99	0074000000000000000143
.62000	.61987	.00013	3	99	0074000000000000000143
.63000	.62982	.00018	3	99	0074000000000000000143
.64000	.63989	.00011	3	99	0074000000000000000143
.65000	.64978	.00022	3	99	0074000000000000000143
.66000	.65985	.00015	3	99	0074000000000000000143
.67000	.66986	.00014	3	99	0074000000000000000143
.68000	.67981	.00019	3	99	0074000000000000000143
.69000	.68988	.00012	3	99	0074000000000000000143
.70000	.69989	.00011	3	99	0074000000000000000143
.71000	.70996	.00004	3	99	0074000000000000000143
.72000	.71991	.00009	3	99	0074000000000000000143
.73000	.72992	.00008	3	99	0074000000000000000143
.74000	.73993	.00007	3	99	0074000000000000000143
.75000	.74994	.00006	3	99	0074000000000000000143
.76000	.75977	.00023	3	99	0074000000000000000143
.77000	.76984	.00016	3	99	0074000000000000000143
.78000	.77979	.00021	3	99	0074000000000000000143
.79000	.78986	.00014	3	99	0074000000000000000143
.80000	.79987	.00013	3	99	0074000000000000000143
.81000	.80994	.00006	3	99	0074000000000000000143
.82000	.81982	.00018	3	99	0074000000000000000143
.83000	.82977	.00023	3	99	0074000000000000000143
.84000	.83984	.00016	3	99	0074000000000000000143
.85000	.84985	.00015	3	99	0074000000000000000143
.86000	.85986	.00014	3	99	0074000000000000000143
.87000	.86993	.00007	3	99	0074000000000000000143
.88000	.87976	.00024	3	99	0074000000000000000143
.89000	.88977	.00023	3	99	0074000000000000000143
.90000	.89984	.00016	3	99	0074000000000000000143
.91000	.90985	.00015	3	99	0074000000000000000143
.92000	.91980	.00020	3	99	0074000000000000000143
.93000	.92987	.00013	3	99	0074000000000000000143
.94000	.93988	.00012	3	99	0074000000000000000143
.95000	.94977	.00023	3	99	0074000000000000000143
.96000	.95984	.00016	3	99	0074000000000000000143
.97000	.96985	.00015	3	99	0074000000000000000143
.98000	.97992	.00008	3	99	0074000000000000000143
.99000	.98993	.00007	3	99	0074000000000000000143
1.00000	.99987	.00013	3	99	0074000000000000000143
1.01000	.99982	.01018	10	98	2074000000000000000142
1.02000	.99982	.02018	10	98	2074000000000000000142
1.03000	.99976	.03024	10	98	2074000000000000000142

1.14000	.99983	.[illegible]12	10	98	20740000[illegible]0[illegible]142
1.15000	.99982	.[illegible]13	10	98	20740000[illegible]0[illegible]142
1.16000	.99982	.05013	10	98	20740[illegible]0[illegible]0[illegible]142
1.17000	.99982	.07013	10	98	20740000[illegible]0[illegible]142
1.18000	.99982	.08013	10	98	20740000000000000142
1.19000	.99976	.09024	10	98	20740000000000000142
1.10000	.99982	.10013	10	98	20740000000000000142
1.11000	.99988	.11012	10	98	20740000000000000142
1.12000	.99988	.12012	10	98	20740000000000000142
1.13000	.99982	.13013	10	98	20740000000000000142
1.14000	.99982	.14013	10	98	20740000000000000142
1.15000	.99982	.15013	10	98	20740000000000000142
1.16000	.99988	.16012	10	98	20740000000000000142
1.17000	.99988	.17012	10	98	20740000000000000142
1.18000	.99982	.18013	10	98	20740000000000000142
1.19000	.99988	.19012	10	98	20740000000000000142
1.20000	.99988	.20012	10	98	20740000000000000142
1.21000	.99982	.21013	10	98	20740000000000000142
1.22000	.99982	.22013	10	98	20740000000000000142
1.23000	.99982	.23013	10	98	20740000000000000142
1.24000	.99982	.24013	10	98	20740000000000000142
1.25000	.99982	.25013	10	98	20740000000000000142
1.26000	.99994	.26006	10	98	30740000000000000142
1.27000	.99988	.27012	10	99	20740000000000000143
1.28000	.99988	.28012	10	98	20740000000000000142
1.29000	.99976	.29024	10	98	20740000000000000142
1.30000	.99988	.30012	10	98	20740000000000000142
1.31000	.99982	.31013	10	98	20740000000000000142
1.32000	.99988	.32012	10	98	20740000000000000142
1.33000	.99976	.33024	10	98	20740000000000000142
1.34000	.99982	.34013	10	98	20740000000000000142
1.35000	.99988	.35012	10	98	20740000000000000142
1.36000	.99982	.36013	10	98	20740000000000000142
1.37000	.99982	.37013	10	98	20740000000000000142
1.38000	.99982	.38013	10	98	20740000000000000142
1.39000	.99988	.39012	10	98	20740000000000000142
1.40000	.99982	.40013	10	98	20740000000000000142
1.41000	.99988	.41012	10	98	20740000000000000142
1.42000	.99988	.42012	10	98	20740000000000000142
1.43000	.99988	.43012	10	98	20740000000000000142
1.44000	.99982	.44013	10	98	20740000000000000142
1.45000	.99988	.45012	10	98	20740000000000000142
1.46000	.99988	.45012	10	98	20740000000000000142
1.47000	.99988	.47012	10	98	20740000000000000142
1.48000	.99988	.48012	10	98	20740000000000000142
1.49000	.99982	.49013	10	98	20740000000000000142
0.00000	-0.00000	0.00000	2	99	00740000000000000143
.01000	.00983	.00017	3	99	00740000000000000143
.02000	.01984	.00016	3	99	00740000000000000143
.03000	.02985	.00015	3	99	00740000000000000143
.04000	.03979	.00021	3	99	00740000000000000143
.05000	.04987	.00013	3	99	00740000000000000143
.06000	.05988	.00012	3	99	00740000000000000143
.07000	.06982	.00018	3	99	00740000000000000143
.08000	.07983	.00017	3	99	00740000000000000143
.09000	.08990	.00010	3	99	00740000000000000143
.10000	.09991	.00009	3	99	00740000000000000143
.11000	.10992	.00008	3	99	00740000000000000143
.12000	.11993	.00007	3	99	00740000000000000143
.13000	.12982	.00018	3	99	00740000000000000143
.14000	.13983	.00017	3	99	00740000000000000143
.15000	.14990	.00010	3	99	00740000000000000143
.16000	.15991	.00009	3	99	00740000000000000143
.17000	.16986	.00014	3	99	00740000000000000143
.18000	.17993	.00007	3	99	00740000000000000143
.19000	.18994	.00006	3	99	00740000000000000143

1.26000	.99982	.26018	10	98	2074000000000000000142
1.27000	.99982	.27018	10	98	2074000000000000000142
1.28000	.99982	.28018	10	98	2074000000000000000142
1.29000	.99982	.29018	10	98	2074000000000000000142
1.30000	.99978	.30024	10	98	2074000000000000000142
1.31000	.99988	.31012	10	98	2074000000000000000142
1.32000	.99982	.32018	10	98	2074000000000000000142
1.33000	.99988	.33012	10	98	2074000000000000000142
1.34000	.99978	.33024	10	98	2074000000000000000142
1.35000	.99982	.34018	10	98	2074000000000000000142
1.36000	.99982	.35018	10	98	2074000000000000000142
1.37000	.99982	.36018	10	98	2074000000000000000142
1.38000	.99982	.37018	10	98	2074000000000000000142
1.39000	.99982	.38018	10	98	2074000000000000000142
1.40000	.99988	.39012	10	98	2074000000000000000142
1.41000	.99982	.40018	10	98	2074000000000000000142
1.42000	.99988	.41012	10	98	2074000000000000000142
1.43000	.99988	.42012	10	98	2074000000000000000142
1.44000	.99982	.43018	10	98	2074000000000000000142
1.45000	.99988	.44012	10	98	2074000000000000000142
1.46000	.99988	.45012	10	98	2074000000000000000142
1.47000	.99988	.46012	10	98	2074000000000000000142
1.48000	.99988	.47012	10	98	2074000000000000000142
1.49000	.99982	.48018	10	98	2074000000000000000142
0.00000	-0.00000	0.00000	2	99	0074000000000000000143
.01000	.00983	.00017	3	99	0074000000000000000143
.02000	.01984	.00016	3	99	0074000000000000000143
.03000	.02985	.00015	3	99	0074000000000000000143
.04000	.03979	.00021	3	99	0074000000000000000143
.05000	.04987	.00013	3	99	0074000000000000000143
.06000	.05988	.00012	3	99	0074000000000000000143
.07000	.06982	.00018	3	99	0074000000000000000143
.08000	.07990	.00010	3	99	0074000000000000000143
.09000	.08990	.00010	3	99	0074000000000000000143
.10000	.09991	.00009	3	99	0074000000000000000143
.11000	.10992	.00008	3	99	0074000000000000000143
.12000	.11993	.00007	3	99	0074000000000000000143
.13000	.12988	.00013	3	99	0074000000000000000143
.14000	.13983	.00017	3	99	0074000000000000000143
.15000	.14990	.00010	3	99	0074000000000000000143
.16000	.15991	.00009	3	99	0074000000000000000143
.17000	.16986	.00014	3	99	0074000000000000000143
.18000	.17983	.00017	3	99	0074000000000000000143
.19000	.18984	.00016	3	99	0074000000000000000143

.22000	.21997	.00003	3	99	0074000000000000000143
.23000	.22993	.00007	3	99	0074000000000000000143
.24000	.23993	.00007	3	99	0074000000000000000143
.25000	.24994	.00006	3	99	0074000000000000000143
.26000	.25989	.00011	3	99	0074000000000000000143
.27000	.26994	.00006	3	99	0074000000000000000143
.28000	.27985	.00015	3	99	0074000000000000000143
.29000	.28979	.00021	3	99	0074000000000000000143
.30000	.29993	.00007	3	99	0074000000000000000143
.31000	.30994	.00006	3	99	0074000000000000000143
.32000	.31989	.00011	3	99	0074000000000000000143
.33000	.32983	.00017	3	99	0074000000000000000143
.34000	.33984	.00016	3	99	0074000000000000000143
.35000	.34991	.00009	3	99	0074000000000000000143
.36000	.35986	.00014	3	99	0074000000000000000143
.37000	.37000	.00000	3	99	0074000000000000000143
.38000	.37988	.00012	3	99	0074000000000000000143
.39000	.38989	.00011	3	99	0074000000000000000143
.40000	.39984	.00016	3	99	0074000000000000000143
.41000	.40997	.00003	3	99	0074000000000000000143
.42000	.41992	.00008	3	99	0074000000000000000143
.43000	.42993	.00007	3	99	0074000000000000000143
.44000	.43976	.00024	3	99	0074000000000000000143
.45000	.44989	.00011	3	99	0074000000000000000143
.46000	.45980	.00018	3	99	0074000000000000000143
.47000	.46991	.00009	3	99	0074000000000000000143
.48000	.47003	.00997	1	99	0074000000000000000143
.49000	.48993	.00007	3	99	0074000000000000000143
.50000	.49005	.00995	2	99	0074000000000000000143
.51000	.50989	.00011	3	99	0074000000000000000143
.52000	.51984	.00016	3	99	0074000000000000000143
.53000	.52985	.00015	3	99	0074000000000000000143
.54000	.53992	.00008	3	99	0074000000000000000143
.55000	.54987	.00013	3	99	0074000000000000000143
.56000	.55981	.00019	3	99	0074000000000000000143
.57000	.56982	.00018	3	99	0074000000000000000143
.58000	.57977	.00023	3	99	0074000000000000000143
.59000	.58984	.00016	3	99	0074000000000000000143
.60000	.59973	.00027	3	99	0074000000000000000143
.61000	.60992	.00008	3	99	0074000000000000000143
.62000	.61987	.00013	3	99	0074000000000000000143
.63000	.62982	.00018	3	99	0074000000000000000143
.64000	.63983	.00017	3	99	0074000000000000000143
.65000	.64984	.00016	3	99	0074000000000000000143
.66000	.65991	.00009	3	99	0074000000000000000143
.67000	.66992	.00008	3	99	0074000000000000000143
.68000	.67993	.00007	3	99	0074000000000000000143
.69000	.68988	.00012	3	99	0074000000000000000143
.70000	.69983	.00017	3	99	0074000000000000000143
.71000	.70984	.00016	3	99	0074000000000000000143
.72000	.71985	.00015	3	99	0074000000000000000143
.73000	.72992	.00008	3	99	0074000000000000000143
.74000	.73993	.00007	3	99	0074000000000000000143
.75000	.74988	.00012	3	99	0074000000000000000143
.76000	.75977	.00023	3	99	0074000000000000000143
.77000	.76984	.00016	3	99	0074000000000000000143
.78000	.77985	.00015	3	99	0074000000000000000143
.79000	.78986	.00014	3	99	0074000000000000000143
.80000	.79987	.00013	3	99	0074000000000000000143
.81000	.80988	.00012	3	99	0074000000000000000143
.82000	.81976	.00024	3	99	0074000000000000000143
.83000	.82983	.00017	3	99	0074000000000000000143
.84000	.83984	.00016	3	99	0074000000000000000143
.85000	.84985	.00015	3	99	0074000000000000000143

.87000	.86484	.00013	3	99	0074000000000000143
.88000	.87982	.00013	3	99	0074000000000000143
.89000	.89477	.00023	3	99	0074000000000000143
.90000	.89984	.00016	3	99	0074000000000000143
.91000	.90981	.00014	3	99	0074000000000000143
.92000	.91985	.00014	3	99	0074000000000000143
.93000	.92987	.00015	3	99	0074000000000000143
.94000	.93982	.00018	3	99	0074000000000000143
.95000	.94983	.00017	3	99	0074000000000000143
.96000	.95984	.00016	3	99	0074000000000000143
.97000	.96981	.00019	3	99	0074000000000000143
.98000	.97980	.00020	3	99	0074000000000000143
.99000	.98983	.00007	3	99	0074000000000000143

PROGRAM TERMINATED ON MAX LINE

APPENDIX F

COMPUTER LISTING OF
PHASE 1 STINGER CONVERSION

```
      PROGRAM MAIN(INPUT,OUTPUT,HFILE,TAPE6=OUTPUT,TAPE5=INPUT,TAPE8)
COMMENT.  ***************
COMMENT.  HYBRID SIMULATION CONTROL
COMMENT.  ***************
C***
C***
C     NRUNS IS THE NUMBER OF RUNS PER FLIGHT CONDITION.  MAXIMUM OF 20
C     XTERM IS THE VALUE OF DELTA X AT WHICH SIMULATION IS TERMINATED
C     DX IS THE DELTA RANGE BETWEEN COLLECTED DATA POINTS
C     NPTS IS THE NUMBER OF DATA POINTS COLLECTED
C***
      COMMON/COMA/LEVEL,IPTS,XXS(50),XDTGO,YDTGO,ZDTGO,RLB,COSE,SPO,RI,
     *GAM,EDOT,THETAL,RN,
     *              PPX(50),PPY(50),PPZ(50),TIME(50),TMAS(30),XDTGMS(30)
     *             ,YDTGMS(30),ZDTGMS(30),XMAN(4,50),XMISS(7),NT
     *             ,XCOMP,YCOMP,ZCOMP,TAMA(30),DELTAR(30),VM(30),G,GGG
     *,XDO,YDO,ZDO,DXG,DYG,DZG,S2,S3,S4,S5,XDM(30),YDM(30),ZDM(30),
     *RLBK,SCALEP,F1,F2,F3,G1,G2,G3,XC,YC,ZC,S1,RRR,SR,SPL,CTL,STL,
     *CPL,A1,VTI,XE(30),YE(30),ZE(30),ZALT,NERR,CLA,NPX,NX,CLAA(10),
     *PHO,ARG,AAA,SCALET,TREAL,TMA(30),XTA,YTA,ZTA,SCALEV,QMM(10),QM
     *,SA,CA,VMX(50),VMY(50),VMZ(50)
     *,ECOS(17),HR7EE(17),FI7T(9),FCI77(9)
      COMMON /COMB/XD,YD,ZD,XDTG,YDTG,ZDTG,XDEF,YDEF,ZDEF,ZZZ
     *,XDDE,YDDE,ZDDE
      COMMON/MANEUV/ITT,XDEM(30,45),YDEM(30,45),ZDEM(30,45),
     *              XDTGM(30),YDTGM(30),ZDTGM(30),XPRIME,YPRIME,
     *              ZPRIME,TIMES(50),ITRAJ(60),ITAU(60),III,NADJ
      COMMON/GTARG/FXA,FYA,FZA,FXB,FYB,FZB,FXC,FYC,FZC,AT,VT
      COMMON/COMP/IS1,I99,I88,CLMS(10),CLS1(10),CLS2(10),NSAM,Q1,Q2,Q3,
     *Q4,Q5,Q6,W1,W2,W3,W4,W5,W6,IFT,IPLANE,ICON,S2P1,S1P1,A2P1,A1P1,
     *S2P2,S1P2,A2P2,A1P2,S2P3,S1P3,A2P3,A1P3,S2P4,S1P4,A2P4,A1P4,S2P5,
     *S1P5,A2P5,A1P5,S2P6,S1P6,A2P6,A1P6
      COMMON/EXTRA/NPTS,DX,XTERM,KKK,AS,SEO,SC,ASEO,R,SCALEL,VVVV,C1,
     *XLOS,YLOS,ZLOS,CO,CX,CY,CLY,CLZ,TRP,XX(50),THE
      COMMON /ANG/ SINSO,COSSO,SINSCO,COSSCO
      COMMON/PK/XDH(22),YDH(22),ZDH(22),BDHX(22),BDHY(22),BDHZ(22)
     *,KILL(22),P(4),SPK,SPHG,SPHS,SPM
      COMMON/MISD/XM(3),YM(3),ZM(3),RM(3),NOPT,ICN
      COMMON/DELT/PPXS(50),PPYS(50),PPZS(50),VMXS(50),VMYS(50),VMZS(50)
      COMMON/COMD/COSA,COSB,COSG,YA,ZA
C***
      REAL   NR
      DIMENSION  ZTEO(62),TM(30)
      DIMENSION  AXM(20),AYM(20),AZM(20),ARM(20),
     *           AXB(20),AYB(20),AZB(20),ARB(20),
     *           AXC(20),AYC(20),AZC(20),ARC(20)
      DIMENSION ASC(62),RR(62),XMT(3),YMT(3),ZMT(3),RMT(3)
      DIMENSION AXMT(20),AYMT(20),AZMT(20),ARMT(20),AXBT(20),AYBT(20),
     *AZBT(20),ARBT(20),AXCT(20),AYCT(20),AZCT(20),ARCT(20)
      INTERRUPT(I=1,R=50,T=80,P=100)
      COMMON/*DAC1/LAUNCH(11)
      COMMON/*ADC1/ADIN(10)
      COMMON/*IDIS2/IIN
      JAM,ON
      COMMON/*ODIS2/IOUT
      JAM,OFF
```

```
C*** NEXT CARD IS TAPE SPECIFIC FOR LAUNCH AT WSMR
      DATA ZABOVE/-3989./
      DATA  RTD,DTR/57.2957795,0.01745329/
      DATA(TM(I),I=1,27)/0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.,12.,
     *       13.,14.,15.,16.,17.,18.,19.,20.,21.,22.,23.,24.,25.,26./
      SCALE = 102.375
      SCALET = 102.375 / 10.
      SCALEL = 200. * 102.375
      SCALEV=20.*102.375
      SCALE2=5.*1.02375
      SCALED=20.*102.375
      KASE=0
      RLBK=3.3333
   10 CONTINUE
      READ (8,2000) IS1,NSAM,ICON,INL
      WRITE (8,800)
      WRITE (8,2002) IS1,NSAM,ICON,INL
      PRINT 800
      DO 11 I=1,10
      READ (5,620) CLMS(I),CLS1(I),CLS2(I)
      PRINT 2005, CLMS(I),CLS1(I),CLS2(I)
   11 CONTINUE
      READ (5,621) Q1,Q2,Q3,Q4,Q5,Q6
      PRINT 800
      PRINT 2006, Q1,Q2,Q3,Q4,Q5,Q6
      READ (5,621) W1,W2,W3,W4,W5,W6
      PRINT 800
      PRINT 2007, W1,W2,W3,W4,W5,W6
      CALL QCLIM
      READ (5,902) NRUNS
      PRINT 800
      PRINT 2010, NRUNS
      NR=NRUNS
      IF (NRUNS.LE.20)  GO TO 15
      PRINT 802, NRUNS
      NRUNS = 20
   15 CONTINUE
      READ (5,903) NPTS,DX,XTERM
      PRINT 800
      PRINT 2008, NPTS,DX,XTERM
C*** READ IN LAUNCH CONDITIONS. NL IS NUMBER OF LAUNCH SETS.
      READ (5,902)  NL
      PRINT 800
      PRINT 2011, NL
      IF(NL.LE.61) GO TO 30
      PRINT 823, NL
      GO TO 3
   30 CONTINUE
      PRINT 800
      IF(IS1.NE.1) PRINT 833
C*** READ AND PRINT LOADED LAUNCH CONDITION SETS.
      NLL=NL+1
      PRINT 800
```

*FOR NL IS TOO LARGE

```
      DO 40 I=1,NLL
      READ (5,904) ZTEO(I),ASC(I),RR(I)
      ZTE = -ZTEO(I)
      IF(IS1.NE.1) PRINT 835, I,ZTE,ASC(I),RR(I)
   40 CONTINUE
    3 CONTINUE
      PRINT 800
      NT=27
      IQC1=0
      GO TO 42
C*** AUTOMATIC SELECTION OF LAUNCH CONDITIONS.
   41 INL=INL+1
      IFT=0
C     INDEX FOR AUTO TEST CASE CHECK
   63 I99=0
      U77=FLOAT(INL)
      U78=(U77-1.)/5.
      I77=IFIX(U78)
      U79=FLOAT(I77)
      U80=U78-U79
C     TEST FOR EVERY FIFTH LAUNCH SET
      IF(U80.LT.0.01) I99=1
C     TEST FOR FIRST LAUNCH SET
      IF(INL.EQ.1) I99=1
      IF(INL.EQ.1) IQC1=1
C     INDEX FOR OTHER RUNS
      I88=0
C     TEST FOR LAST TEST CASE CHECK
      IF(INL.GT.NL) I88=1
C     TEST CASE CHECK REQUIRED AFTER LAST LAUNCH SET
      IF(INL.GT.NL) I99=1
      IF(I99.EQ.1) GO TO 62
  628 KASE = INL
      KA=KASE
      PRINT 918, KA
      ITT=ITRAJ(INL)
      PRINT 916,ITT
      IT1=ITAU(INL)
      PRINT 917,IT1
      TAU=FLOAT(IT1)
      GO TO 51
   42 READ (8,2000) III,IRSS,KA,ITT,NOGO
      WRITE (8,800)
      WRITE (8,2003) III,IRSS,KA,ITT,NOGO
      READ (8,2001) TAU
      WRITE (8,800)
      WRITE (8,2004) TAU
      IF(III.EQ.1) GO TO 41
   50 CONTINUE
    5 PRINT 800
   23 CONTINUE
C*** LAUNCH CONDITIONS HAVE BEEN SELECTED.
      GO TO 51
   62 CONTINUE
      PRINT 625
      TAU=3.0
C     INDEX FOR TEST CASE DATA CARD OF LAUNCH CONDITIONS
```

```
      KA=NL+1
C     INDEX FOR TEST CASE TARGET MANEUVER
C*** NEXT CARD IS TAPE SPECIFIC FOR ITT=39 FOR TEST CASE MANEUVER
      ITT=39
C*** ADJUST FOR TAU AND SCALE TARGET VELOCITY-TIME MANEUVER TABLE.
   51 NTAU = INT(TAU) + 1
      DO 25 I = NTAU,NT
      K = I - NTAU + 1
      TMA(K)=TM(I)-TAU
      TMAS(K)=TMA(K)/SCALET
      XDM(K)=XDEM(I,ITT)
      YDM(K)=YDEM(I,ITT)
      ZDM(K)=ZDEM(I,ITT)
   25 CONTINUE
      KK=NT-INT(TAU)+1
      DO 26 K=KK,NT
      TMA(K)=FLOAT(K)-1.
      XDM(K)=XDEM(NT,ITT)
      YDM(K)=YDEM(NT,ITT)
      ZDM(K)=ZDEM(NT,ITT)
   26 CONTINUE
C*** GENERATE TARGET AERODYNAMIC ACCELERATION TABLE
      XE(1)=XDM(2)-XDM(1)
      YE(1)=YDM(2)-YDM(1)
      ZE(1)=ZDM(2)-ZDM(1)-32.174
      DO 31 I=2,NT
      IV=I-1
      XE(I)=XDM(I)-XDM(IV)
      YE(I)=YDM(I)-YDM(IV)
      ZE(I)=ZDM(I)-ZDM(IV)-32.174
   31 CONTINUE
      KASE=KA
      Z=ZTEO(KASE)
      ZZZ=Z+ZABOVE
      ZALT=ZZZ
      AS=ASC(KASE)
      R=RR(KASE)
      XD=XDM(1)
      YD=YDM(1)
      ZD=ZDM(1)
C     CHECK FOR HELICOPTER
      VVVV=SQRT(XD*XD+YD*YD+ZD*ZD)
      IF(VVVV.LT.338.0) PRINT 199
C***
      KKK=0
      IF(XD.NE.0.0.AND.YD.NE.0.0) GO TO 55
      IF(XD.EQ.0.0.AND.YD.GT.0.0) GO TO 53
      IF(XD.EQ.0.0.AND.YD.LT.0.0) GO TO 54
      KKK=1
      SCO=0.0
      ASCO=0.0
      IF(ZD.EQ.0.0) GO TO 52
      IF(ZD.GT.0.0) SEO=90.*DTR
      IF(ZD.LT.0.0) SEO=-90.*DTR
      PRINT 913
      GO TO 60
   52 PRINT 914
```

```
      SEO=0.0
      GO TO 60
   53 BETA=90.
      GO TO 56
   54 BETA=270.
      GO TO 56
   55 BETA=ATAN2(YD,XD)*RTD
   56 ASCO=AS+BETA-180.
      IF(ASCO.GT.0.0) GO TO 60
      ASCO=360.+ASCO
   60 CONTINUE
      SCO=ASCO*DTR
      ARG=-Z/R
      SO=ASIN(ARG)
      THETAL=SO*RTD/SCALE
      ASO=SO*RTD
      X=R*COS(SO)*COS(SCO)
      Y=R*COS(SO)*SIN(SCO)
      PRINT 805, ASCO,ASO,R
      PRINT 912,X,Y,Z
C*** CALCULATE CONSTANTS FOR CORDINATE TRANSFORMATION
      SINSO=SIN(SO)
      COSSO=COS(SO)
      SINSCO=SIN(SCO)
      COSSCO=COS(SCO)
      S2=COSSO*COSSCO
      S3=SINSO*COSSCO
      S4=COSSO*SINSCO
      S5=SINSO*SINSCO
      SPL=SINSCO
      CTL=COSSO
      STL=SINSO
      CPL=COSSCO
C*** MAKE EARTH TO GUIDANCE TRANSFORMATION
      CALL GUID
      XDTGO = XDTG
      YDTGO = YDTG
      ZDTGO = ZDTG
C*** CALCULATE SCALE FACTOR EQUATION
      RI=R
      CALL KSCALE,RETURNS(50)
      EDOT=(GGG/G)/SCALE2
C*** READY TO MAKE FLIGHTS AND COLLECT POSITION POINTS
      SPK=0.
      SPHG=0.
      SPHS=0.
      SPM=0.
      N111=NRUNS
      IF(IRSS.EQ.1) NRUNS=IRSS
      IF(IRSS.EQ.1) NR=IRSS
      IF(IRSS.EQ.1) I99=0
      IF(I99.EQ.1) NRUNS=NSAM+1
      IF(I99.EQ.1) NR=NRUNS
      DO 150 I=1,NRUNS
      CALL INIT
      IF(IS1.NE.1) PRINT 77,RN
      ICN = I
```

```
      XMISS(7) = -77
C*** ZERO STORAGES AREA
      DO 85 J=1,50
      DO 85 K=1,4
      XMAN(K,J) = 0.
   85 CONTINUE
      DO 90 J=1,50
      PPX(J) = 0.0
      PPY(J) = 0.0
      PPZ(J) = 0.0
      TIME(J) = 0.0
      VMX(J) = 0.0
      VMY(J) = 0.0
      VMZ(J) = 0.0
   90 CONTINUE
      IPTS=NADJ
      CALL FLIGHT
      IF(IS1.EQ.1) GO TO 91
      PRINT 999,RI,GAM,EDOT,XDTGO,YDTGO,ZDTGO,RN,RLB,SPD
      PRINT 999,XCOMP,YCOMP,ZCOMP,G,GGG,XDO,YDO,ZDO,DXG,DYG
      PRINT 999,DZG,S2,S3,S4,S5,RLBK,SCALEP,COSE,F1,F2
      PRINT 999,F3,G1,G2,G3,XC,YC,ZC,S1,RRR,SR
      PRINT 999,SPL,CTL,STL,CPL,A1,VTI,ZALT,CLA,PHO,ARG
      PRINT 999,AAA,SCALET,TREAL,XTA,YTA,ZTA,SCALEV,QM,SA,CA
      PRINT 996,LEVEL,IPTS,NT,NERR,NPX,NX
      PRINT 999,TRP,XLOS,YLOS,ZLOS,CO,CX,CY,CLY,CLZ,VVVV
      PRINT 999,C1,AS,SEO,SC,ASEO,R,SCALEL,THETAL,THE
   91 CONTINUE
      NOPT = NADJ
C*** SCALE AND WRITE THE POSITION POINTS
      IF(IS1.NE.1) PRINT 800
      IF (XMISS(7).GE.0.)  PRINT 909, XMISS
      IF(IS1.EQ.1) PRINT 809,ICN
      IF(IS1.NE.1) PRINT 810,ICN
   95 IF (IPTS.EQ.0) GO TO 150
      DO 110 J=1,NADJ
      PPXS(J) = PPX(J) * SCALEP
      PPYS(J) = PPY(J) * SCALEP
      PPZS(J) = PPZ(J) * SCALEP
      TIMES(J)=TIME(J) * SCALET
      VMXS(J)=VMX(J)*SCALEL
      VMYS(J)=VMY(J)*SCALEL
      VMZS(J)=VMZ(J)*SCALEL
      IF(IS1.EQ.1) GO TO 110
      PRINT 811, PPXS(J),PPYS(J),PPZS(J),TIMES(J)
  110 CONTINUE
C     CHECK DIGITAL TIME CRITICAL DATA COLLECTION
      NOK=NADJ-1
      DO 330 M=1,NOK
      XCOMM=-XX(M)
      XCOLL=PPXS(M)
      XCODX=-XX(M+1)
      IF(XCOMM.LT.XCOLL.AND.XCOLL.LT.XCODX) GO TO 330
      GO TO 329
  330 CONTINUE
      XCOMM=-XX(NADJ)
      XCOLL=PPXS(NADJ)
```

ICN flight number

```
      XCODX=-XX(NADJ)+DX
      IF(XCOMM.LT.XCOLL.AND.XCOLL.LT.XCODX) GO TO 83
  329 DO 331 M=1,20
  331 PRINT 332
   83 CONTINUE
      IF (LEVEL.LT.0)  GO TO 116
C*** SCALE AND WRITE THE MANEUVER DATA
      IF(IS1.EQ.1) GO TO 111
      PRINT 907
  111 CONTINUE
      DO 114 J=1,NADJ
      XMAN(1,J) = XMAN(1,J) * SCALET
      DO 112 K=2,4
      XMAN(K,J) = XMAN(K,J) * SCALEV
  112 CONTINUE
      IF(IS1.EQ.1) GO TO 114
      PRINT 908, (XMAN(K,J),K=1,4)
  114 CONTINUE
      IF(IS1.NE.1) PRINT 919,XDO,YDO,ZDO
  116 CONTINUE
C*** NEXT CARD IS TAPE SPECIFIC FOR ITT=1 FOR STATIONARY TARGET
      IF(ITT.NE.1.AND.LEVEL.EQ.0.AND.VVVV.GE.338.0) CALL PRIME
      IF(LEVEL.NE.0) GO TO 150
C*** CALCULATE THE MISS-DISTANCE
      CALL MISCOM
      PRINT 815
      PRINT 813
      DO 120 K=1,3
      PRINT 814, XM(K),YM(K),ZM(K),RM(K)
  120 CONTINUE
C*** SAVE THE MISS-DISTANCE DATA
      AXM(ICN) = XM(1)
      AYM(ICN) = YM(1)
      AZM(ICN) = ZM(1)
      ARM(ICN) = RM(1)
      AXB(ICN) = XM(2)
      AYB(ICN) = YM(2)
      AZB(ICN) = ZM(2)
      ARB(ICN) = RM(2)
      AXC(ICN) = XM(3)
      AYC(ICN) = YM(3)
      AZC(ICN) = ZM(3)
      ARC(ICN) = RM(3)
      IF(VVVV.LT.338.0) GO TO 150
      XMT(1)=XM(1)
      YMT(1)=YM(1)+19.5*YA
      ZMT(1)=ZM(1)+19.5*ZA
      XMT(2)=XM(2)-19.5
      YMT(2)=YM(2)
      ZMT(2)=ZM(2)
      XMT(3)=XM(3)-19.5
      YMT(3)=YM(3)
      ZMT(3)=ZM(3)
      AXMT(ICN)=XMT(1)
      AYMT(ICN)=YMT(1)
      AZMT(ICN)=ZMT(1)
      AXBT(ICN)=XMT(2)
```

REFERENCED TO TAILPIPE

xm	ym	zm	RM
xm(1)	ym(1)	zm(1)	RM(1)
xm(2)	ym(2)	zm(2)	RM(2)
xm(3)	xm		

22 ft CG

```
      AYBT(ICN)=YMT(2)
      AZBT(ICN)=ZMT(2)
      AXCT(ICN)=XMT(3)
      AYCT(ICN)=YMT(3)
      AZCT(ICN)=ZMT(3)
      DO 121 K=1,3
  121 RMT(K)=SQRT(XMT(K)**2+YMT(K)**2+ZMT(K)**2)
      PRINT 800
      ARMT(ICN)=RMT(1)
      ARBT(ICN)=RMT(2)
      ARCT(ICN)=RMT(3)
      PRINT 816
      PRINT 813
      DO 122 K=1,3
      PRINT 814, XMT(K),YMT(K),ZMT(K),RMT(K)
  122 CONTINUE
      CALL LETH
C*** GO BACK FOR THE NEXT FLIGHT
  150 CONTINUE
C*** AFTER FLIGHTS HAVE BEEN COMPLETED, CALCULATE MEAN AND STD DEV
      PRINT 815
      PRINT 830
      IPLANE=0
      CALL MEAN (AXM,AYM,AZM,ARM,NRUNS,XM(1),YM(1),ZM(1),RM(1))
      PRINT 831
      CALL MEAN (AXB,AYB,AZB,ARB,NRUNS,XM(2),YM(2),ZM(2),RM(2))
      PRINT 832
      CALL MEAN (AXC,AYC,AZC,ARC,NRUNS,XM(3),YM(3),ZM(3),RM(3))
C*** GO BACK FOR THE NEXT SET OF FLIGHT CONDITIONS
      IF(VVVV.LT.338.0) GO TO 197
      PRINT 800
      PRINT 816
      PRINT 830
      IF(I99.EQ.1) IPLANE = 1
      CALL MEAN (AXMT,AYMT,AZMT,ARMT,NRUNS,XMT(1),YMT(1),ZMT(1),RMT(1))
      PRINT 831
      IF(I99.EQ.1) IPLANE = 2
      CALL MEAN (AXBT,AYBT,AZBT,ARBT,NRUNS,XMT(2),YMT(2),ZMT(2),RMT(2))
      PRINT 832
      IF(I99.EQ.1) IPLANE = 3
      CALL MEAN (AXCT,AYCT,AZCT,ARCT,NRUNS,XMT(3),YMT(3),ZMT(3),RMT(3))
      AARUNS=SPK/NR
      AASUM=SPHG/NR
      ABSUM=SPHS/NR
      ABRUNS=SPM/NR
      PRINT 102,ABRUNS,ABSUM,AASUM,AARUNS
  197 CONTINUE
      NRUNS = N111
      NR = N111
      IF(IFT.NE.1) GO TO 602
      DO 630 I=1,10
  630 PRINT 601
      IFT=0
      IF(ICON.EQ.1) GO TO 635
  609 PAUSE
      IF(IQC1.EQ.1) GO TO 62
      IF(I88.EQ.1) GO TO 240
```

```
      INL=INL-4
      GO TO 62
  240 INNCL=NL
  241 INNCL=INNCL-5
      IF(INNCL.GT.5) GO TO 241
      I88=0
      INL=INL-INNCL
      GO TO 62
  602 IF(I99.NE.1) GO TO 627
      IF(IQC1.EQ.1) GO TO 626
      IF(I88.EQ.1) GO TO 50
      GO TO 633
  626 IQC1=0
      GO TO 633
  627 IF(IRSS.NE.1) GO TO 50
      GO TO 41
  635 IQC1=0
      IF(I88.EQ.1) GO TO 50
  633 I99=0
      GO TO 628
  198 GO TO 50
  200 CONTINUE
      IF (NOGO.EQ.0) GO TO 300
      GO TO 3
  201 PRINT 840
   77 FORMAT(1H0,10X,17H RANDOM NUMBER IS,F10.4)
  102 FORMAT(1H0,8H       =,F6.4,3X,8H       =,F6.4,3X,8H       =,F6.4,
     *3X,8H       =,F6.4)
  199 FORMAT(10X,37H TARGET IS ASSUMED TO BE A HELICOPTER)
  332 FORMAT(39H TIME CRITICAL DATA COLLECTION FAILURE.)
  600 FORMAT(10X,51H IF AUTOMATIC MODE SELECTION IS PLANNED, SELECT THE
     *           46H DESIRED STARTING LAUNCH SET NUMBER WITH SENSE
     *           10H SWITCHES./
     *       10X,52H IF MANUAL MODE SELECTION IS PLANNED, SENSE SWITCHES
     *           31H WILL BE IGNORED AT THIS POINT./
     *       10X,12H PUSH START.)
  601 FORMAT(24H TEST CASE OUT OF LIMITS //)
  605 FORMAT(10X,49H IF COMPLETE PRINTOUT IS DESIRED, SELECT  1  WITH
     *           16H SENSE SWITCHES./
     *       10X,51H IF ONLY ESSENTIAL PRINTOUT IS DESIRED, SELECT  0
     *           20HWITH SENSE SWITCHES./
     *       10X,12H PUSH START.)
  620 FORMAT(3F6.3)
  621 FORMAT(6F10.4)
  622 FORMAT(10X,49H SELECT THE QUALITY CONTROL TEST CASE SAMPLE SIZE
     *           21H WITH SENSE SWITCHES./
     *       10X,43H SAMPLE SIZE MUST BE IN RANGE 2 THROUGH 11.)
  623 FORMAT(10X,50H IF AUTOMATIC MODE SELECTION IS PLANNED, AND IT IS
     *           49H DESIRED TO CONTINUE IF A QUALITY CONTROL FAILURE
     *            8H OCCURS,/
     *       10X,48H THEN SELECT  1  WITH SENSE SWITCHES. OTHERWISE,
     *           13H SELECT  2  .)
  625 FORMAT(//10X,16H TEST CASE ...../)
  640 FORMAT(10X,43H IF THIS IS A MICOM HYBRID/IRSS RUN, SELECT
     *           23H 1 WITH SENSE SWITCHES./
     *       10X,44H IF IT IS NOT, SELECT 2 WITH SENSE SWITCHES./
     *       10X,12H PUSH START.)
```

```
  710 FORMAT(10X,46H SELECT TARGET TRAJECTORY WITH SENSE SWITCHES./
     *       10X,18H THEN PRESS START.)
  715 FORMAT(10X,20H TARGET TRAJECTORY  ,I2,15H  WAS SELECTED./
     *       10X,32H IF THIS IS CORRECT, PUSH START./
     *       10X,36H IF THIS IS WRONG, TURN SWITCHES OFF
     *           16H AND PUSH START.)
  720 FORMAT(10X,49H SELECT POINT TAU ON TARGET TRAJECTORY AT TIME OF
     *           16H MISSILE LAUNCH./10X,20H USE SENSE SWITCHES.)
  725 FORMAT(10X,6H TAU = ,I2,39H  WAS SELECTED. TO CONTINUE, SELECT  1
     *           41H WITH SENSE SWITCH IF TAU=0 WAS SELECTED./
     *       10X,52H IF TAU=0 WAS NOT SELECTED, LEAVE SENSE SWITCHES AS
     *            9HTHEY ARE./
     *       10X,12H PUSH START.)
  800 FORMAT (1H0)
  801 FORMAT(10X 50H SELECT THE LAUNCH CONDITIONS WITH SENSE SWITCHES./
     *        10X 18H THEN PRESS START./
     *        10X 50H TO END RUN, TURN ALL SWITCHES ON AND PRESS START )
  802 FORMAT (10X I5, 39H IS TOO MANY RUNS. IT IS CHANGED TO 20 )
  803 FORMAT(10X,18H LAUNCH CONDITIONS,I5,4X,10H WERE SET./
     *        10X 32H IF THIS IS CORRECT, PUSH START.        /
     *        10X 36H IF THIS IS WRONG, TURN SWITCHES OFF
     *           16H AND PUSH START.)
  805 FORMAT(10X,13H PSI LAUNCH =F7.2,4H DEG,5X,15H THETA LAUNCH =F6.2,
     *4H DEG,5X,16H INITIAL RANGE =F9.2,3H FT)
  809 FORMAT(10X,11H FLIGHT NO. I5)
  810 FORMAT (5X 11H FLIGHT NO. I5,20X 1HX 19X 1HY 19X 1HZ 18X 5H TIME)
  811 FORMAT(26X  4(5X F15.4))
  813 FORMAT (10X 14H                  8X 3H XM 12X 3H YM 12X 3H ZM
     *        12X  3H RM )
  814 FORMAT (24X 3(5X F10.2),E15.8)
  815 FORMAT(10X,23H REFERENCED TO TAILPIPE)
  816 FORMAT(10X,17H REFERENCED TO CG)
  823 FORMAT (5X  I5,5X 20H FOR NL IS TOO LARGE )
  830 FORMAT (1H0 10X 20H Y-Z PLANE - - - - - )
  831 FORMAT (1H0 10X 20H X-Y PLANE - - - - - )
  832 FORMAT (1H0 10X 20H X-Z PLANE - - - - - )
  833 FORMAT (10X 27H TABLE OF LAUNCH CONDITIONS ///)
  835 FORMAT(10X,5H SET I2,5X,11H ALTITUDE =F9.2,2X,3H FT,2X,
     *17H CROSSING ANGLE =F7.2,2X,4H DEG,2X,16H INITIAL RANGE =F10.2,
     *2X,3H FT)
  840 FORMAT (10X 26H END OF RUN.  SIGNING OFF. )
  902 FORMAT(I5)
  903 FORMAT(I10,2F10.0)
  904 FORMAT(3F10.2)
  907 FORMAT (1H0 4X 35H TIME    XD MANEUVER    YD MANEUVER
     *              15H    ZD MANEUVER)
  908 FORMAT(1X F10.4, 3(3X F8.2,4X))
  909 FORMAT (5H X = F7.4,2X 5H Y = F7.4,2X 5H Z = F7.4,2X 8H XDOT =
     *       F7.4,2X 8H YDOT = F7.4,2X 8H ZDOT = F7.4,8H  MISS= E10.3)
  912 FORMAT(10X,7H XTEO =F10.2,3H FT,5X,7H YTEO =F10.2,3H FT,5X,
     *7H ZTEO =F10.2,3H FT)
  913 FORMAT(10X,62H VERTICAL TARGET TRAJECTORY. CROSSING ANGLE IS NOT A
     *PPLICABLE.)
  914 FORMAT(10X,53H STATIONARY TARGET. CROSSING ANGLE IS NOT APPLICABLE
     *.)
  915 FORMAT(10X,94H PUSH ALL SENSE SWITCHES ON FOR AUTOMATIC OPERATION
     *OR ELSE MANUAL OPERATION WILL BE SELECTED./10X,12H PUSH START.)
```

```
  916 FORMAT(10X,19H TARGET TRAJECTORY ,I2)
  917 FORMAT(10X,7H TAU = ,I2)
  918 FORMAT(////10X,5H SET ,I2)
  919 FORMAT(10X,5HXDO =,F10.4,5X,5HYDO =,F10.4,5X,5HZDO =,F10.4)
  996 FORMAT(6I5)
  999 FORMAT(10(1X,E11.4))
 2000 FORMAT(8I5)
 2001 FORMAT(F10.0)
 2002 FORMAT (5X,6HIS1 = ,I5,5X,7HNSAM = ,I5,5X,7HICON = ,I5,5X,6HINL =
     1,I5)
 2003 FORMAT (5X,6HIII = ,I5,5X,7HIRSS = ,I5,5X,5HKA = ,I5,5X,6HITT = ,I
     15,5X,7HNOGO = ,I5)
 2004 FORMAT (5X,6HTAU = ,F10.5)
 2005 FORMAT (5X,7HCLMS = ,F10.5,5X,7HCLS1 = ,F10.5,5X,7HCLS2 = ,F10.5)
 2006 FORMAT (5X,5HQ1 = ,F10.5,5X,5HQ2 = ,F10.5,5X,5HQ3 = ,F10.5,5X,5HQ4
     1 = ,F10.5,5X,5HQ5 = ,F10.5,5X,5HQ6 = ,F10.5)
 2007 FORMAT (5X,5HW1 = ,F10.5,5X,5HW2 = ,F10.5,5X,5HW3 = ,F10.5,5X,5HW4
     1 = ,F10.5,5X,5HW5 = ,F10.5,5X,5HW6 = ,F10.5)
 2008 FORMAT (5X,7HNPTS = ,I5,5X,5HDX = ,F10.5,5X,8HXTERM = ,F10.5)
 2009 FORMAT (5X,7HZTEO = ,F10.5,5X,6HASC = ,F10.5,5X,5HRR = ,F10.5)
 2010 FORMAT (5X,8HNRUNS = ,I5)
 2011 FORMAT (5X,5HNL = ,I5)
  300 STOP
      END
```

```
      BLOCK DATA ONE
      COMMON/MANEUV/ITT,XDEM(30,45),YDEM(30,45),ZDEM(30,45),
     *              XDTGM(30),YDTGM(30),ZDTGM(30),XPRIME,YPRIME,
     *             ZPRIME,TIMES(50),ITRAJ(60),ITAU(60),III,NADJ
      DATA(ITRAJ(I),I=1,59)/3*4,5,3*6,7,8,9,10,11,12,2*13,14,2*15,16,    SERIES
     *17,2*18,12,19,20,21,22,23,24,25,21,26,27,28,29,30,31,21,32,33,34, SERIES
     *35,36,37,38,39,40,41,39,42,38,43,34,44,5*45/                      SERIES
      DATA(ITAU(I),I=1,59)/59*0/                                        SERIES
      DATA(XDEM(I,1),I=1,27)/27*0.0/                                    II
      DATA(YDEM(I,1),I=1,27)/27*0.0/
      DATA(ZDEM(I,1),I=1,27)/27*0.0/
      DATA(XDEM(I,2),I=1,27)/27*0.0/                                    JJ
      DATA(YDEM(I,2),I=1,27)/27*168.9/
      DATA(ZDEM(I,2),I=1,27)/27*0.0/
      DATA(XDEM(I,3),I=1,27)/27*0.0/                                    KK
      DATA(YDEM(I,3),I=1,27)/27*337.8/
      DATA(ZDEM(I,3),I=1,27)/27*0.0/
      DATA(XDEM(I,4),I=1,27)/27*0.0/                                    LL
      DATA(YDEM(I,4),I=1,27)/27*506.7/
      DATA(ZDEM(I,4),I=1,27)/27*0.0/
      DATA(XDEM(I,5),I=1,27)/0.,127.9,251.2,365.4,466.4,550.5,614.7,    A
     *656.6,674.8,668.6,638.2,584.7,510.,416.9,308.7,189.4,63.2,-65.3,
     *-191.5,-310.7,-418.7,-511.5,-585.8,-638.9,-668.9,-674.7,-656.1/
      DATA(YDEM(I,5),I=1,27)/675.6,663.4,627.1,568.2,488.8,391.6,280.3,
     *158.8,31.6,-96.7,-221.6,-338.4,-443.,-531.6,-600.9,-648.5,-672.6,
     *-672.4,-647.9,-599.9,-530.2,-441.3,-336.5,-219.5,-94.6,33.8,161./
      DATA(ZDEM(I,5),I=1,27)/27*0.0/
      DATA(XDEM(I,6),I=1,27)/27*0.0/                                    MM
      DATA(YDEM(I,6),I=1,27)/27*675.6/
      DATA(ZDEM(I,6),I=1,27)/27*0.0/
      DATA(XDEM(I,7),I=1,27)/0.,-221.1,-417.8,-568.5,-656.6,-672.5,     B
     *-614.2,-488.4,-308.7,-95.1,129.,338.9,511.5,627.7,674.9,647.7,
     *549.2,390.3,188.3,-34.4,-253.2,-444.3,-586.3,-663.9,-668.3,-599.1,
     *-464./
      DATA(YDEM(I,7),I=1,27)/634.8,599.9,498.9,342.9,149.2,-60.9,-264.3,
     *-438.6,-564.7,-628.5,-623.2,-549.2,-414.7,-234.6,-28.7,180.4,
     *369.7,518.2,609.7,634.,588.5,478.3,315.3,117.7,-92.9,-293.3,
     *-461.4/
      DATA(ZDEM(I,7),I=1,27)/231.1,218.3,181.6,124.8,54.3,-22.2,-96.2,
     *-159.7,-205.4,-228.8,-226.8,-199.9,-151.,-85.4,-10.4,65.7,134.5,
     *188.6,221.9,230.8,214.2,174.1,114.8,42.8,-33.8,-106.8,-167.9/
      DATA(XDEM(I,8),I=1,27)/0.,-127.9,-251.2,-365.4,-466.4,-550.5,     C
     *-614.7,-656.6,-674.8,-668.6,-638.2,-584.7,-510.,-416.9,-308.7,
     *-189.4,-63.2,65.3,191.5,310.7,418.7,511.5,585.8,638.9,668.9,674.7,
     *656.1/
      DATA(YDEM(I,8),I=1,27)/675.6,663.4,627.1,568.2,488.8,391.6,280.3,
     *158.8,31.6,-96.7,-221.6,-338.4,-443.,-531.6,-600.9,-648.5,-672.6,
     *-672.4,-647.9,-599.9,-530.2,-441.3,-336.5,-219.5,-94.6,33.8,161./
      DATA(ZDEM(I,8),I=1,27)/27*0.0/
      DATA(XDEM(I,9),I=1,27)/27*0.0/                                    NN
      DATA(YDEM(I,9),I=1,27)/27*634.8/
      DATA(ZDEM(I,9),I=1,27)/27*231.1/
      DATA(XDEM(I,10),I=1,27)/27*0.0/                                   OO
      DATA(YDEM(I,10),I=1,27)/27*714.2/
      DATA(ZDEM(I,10),I=1,27)/27*259.9/
      DATA(XDEM(I,11),I=1,27)/27*0.0/                                   D
      DATA(YDEM(I,11),I=1,27)/844.5,822.5,757.7,653.5,515.4,350.4,167.1,
```

```
     *-24.8,-215.4,-394.9,-553.8,-683.8,-778.3,-832.3,-843.,-809.8,
     *-734.5,-621.,-475.2,-304.6,-118.2,74.4,263.,438.,590.3,711.8,
     *796.2/
      DATA(ZDEM(I,11),I=1,27)/0.,-191.4,-372.8,-534.8,-669.,-768.4,
     *-827.8,-844.1,-816.5,-746.4,-637.6,-495.5,-327.6,-142.7,49.6,
     *239.4,416.6,572.3,698.1,787.6,836.2,841.2,802.4,722.,603.9,454.4,
     *281.3/
      DATA(XDEM(I,12),I=1,27)/27*0.0/                                    Q
      DATA(YDEM(I,12),I=1,27)/815.7,844.,828.4,769.7,670.9,537.3,375.7,
     *194.5,3.2,-188.2,-369.9,-532.3,-667.,-767.,-827.1,-844.2,-817.3,
     *-748.,-639.7,-498.1,-330.6,-145.9,46.4,236.3,413.8,569.9,696.3/
      DATA(ZDEM(I,12),I=1,27)/218.6,28.,-164.,-347.4,-512.8,-651.5,
     *-756.3,-821.8,-844.5,-823.2,-759.2,-655.6,-517.9,-353.3,-170.3,
     *21.6,212.3,392.,551.3,681.9,777.1,831.8,843.2,810.7,736.1,623.2,
     *477.8/
      DATA(XDEM(I,13),I=1,27)/0.,159.9,314.,456.8,583.,688.1,768.4,      E
     *820.8,843.5,835.8,797.8,730.9,637.6,521.2,385.9,236.7,78.9,-81.7,
     *-239.4,-388.4,-523.3,-639.4,-732.2,-798.6,-836.2,-843.4,-820.2/
      DATA(YDEM(I,13),I=1,27)/844.5,829.2,783.9,710.3,610.9,489.5,350.4,
     *198.5,39.5,-120.9,-277.,-423.,-553.8,-664.5,-751.1,-810.6,-840.8,
     *-840.5,-809.8,-749.9,-662.8,-551.7,-420.6,-274.4,-118.2,42.3,
     *201.2/
      DATA(ZDEM(I,13),I=1,27)/27*0.0/
      DATA(XDEM(I,14),I=1,27)/0.,-64.3,-128.2,-191.4,-253.4,-314.,       F
     *-372.8,-429.4,-483.5,-534.8,-583.,-627.8,-669.,-706.3,-739.5,
     *-768.4,-792.8,-812.6,-827.8,-838.1,-843.5,-844.1,-839.8,-830.6,
     *-816.5,-797.8,-774.4/
      DATA(YDEM(I,14),I=1,27)/793.5,791.2,784.3,772.9,757.,736.6,712.,
     *683.3,650.6,614.1,574.1,530.7,484.3,435.,383.2,329.2,273.3,215.8,
     *157.,97.4,37.1,-23.3,-83.6,-143.5,-202.5,-260.3,-316.6/
      DATA(ZDEM(I,14),I=1,27)/288.8,288.,285.5,281.3,275.5,268.1,259.2,
     *248.7,236.8,223.5,209.,193.2,176.3,158.3,139.5,119.8,99.5,78.5,
     *57.2,35.4,13.5,-8.5,-30.4,-52.2,-73.7,-94.7,-115.2/
      DATA(XDEM(I,15),I=1,27)/27*0.0/                                    G
      DATA(YDEM(I,15),I=1,27)/844.5,814.6,727.2,588.3,407.8,198.5,-24.8,
     *-246.4,-450.6,-622.9,-751.1,-826.3,-843.0,-800.1,-700.7,-551.7,
     *-363.7,-150.,74.4,293.4,491.7,655.3,772.5,835.1,838.6,782.9,671.8/
      DATA(ZDEM(I,15),I=1,27)/0.,-222.6,-429.4,-605.8,-739.5,-820.8,
     *-844.1,-807.7,-714.2,-570.2,-385.9,-174.3,49.6,270.,471.4,639.4,
     *762.1,831.,841.2,791.9,686.5,532.7,341.1,125.5,-99.,-316.6,-511.7/
      DATA(XDEM(I,16),I=1,27)/0.,64.3,128.2,191.4,253.4,314.,372.8,      H
     *429.4,483.5,534.8,583.,627.8,669.,706.3,739.5,768.4,792.8,812.6,
     *827.8,838.1,843.5,844.1,839.8,830.6,816.5,797.8,774.4/
      DATA(YDEM(I,16),I=1,27)/793.5,791.2,784.3,772.9,757.,736.6,712.,
     *683.3,650.6,614.1,574.1,530.7,484.3,435.,383.2,329.2,273.3,215.8,
     *157.,97.4,37.1,-23.3,-83.6,-143.5,-202.5,-260.3,-316.6/
      DATA(ZDEM(I,16),I=1,27)/288.8,288.,285.5,281.3,275.5,268.1,259.2,
     *248.7,236.8,223.5,209.,193.2,176.3,158.3,139.5,119.8,99.5,78.5,
     *57.2,35.4,13.5,-8.5,-30.4,-52.2,-73.7,-94.7,-115.5/
      DATA(XDEM(I,17),I=1,27)/27*0.0/                                    I
      DATA(YDEM(I,17),I=1,27)/844.5,839.,822.5,795.3,757.7,710.3,653.5,
     *588.3,515.4,435.7,350.4,260.4,167.1,71.6,-24.8,-120.9,-215.5,
     *-307.2,-394.9,-477.4,-553.8,-622.9,-683.8,-735.9,-778.3,-810.6,
     *-832.3/
      DATA(ZDEM(I,17),I=1,27)/0.,-96.3,-191.4,-283.9,-372.8,-456.8,
     *-534.8,-605.8,-669.,-723.4,-768.4,-803.3,-827.8,-841.4,-844.1,
     *-835.8,-816.5,-786.6,-746.5,-696.5,-637.6,-570.2,-495.5,-414.3,
```

```
     *-327.6,-236.7,-142.7/
      DATA(XDEM(I,18),I=1,27)/0.,-159.9,-314.,-456.8,-583.,-688.1,        J
     *-768.4,-820.8,-843.5,-835.8,-797.8,-730.9,-637.6,-521.2,-385.9,
     *-236.7,-78.9,81.7,239.4,388.4,523.3,639.4,732.2,798.6,836.2,843.4,
     *820.2/
      DATA(YDEM(I,18),I=1,27)/844.5,829.2,783.9,710.3,610.9,489.5,350.4,
     *198.5,39.5,-120.9,-277.,-423.,-553.8,-664.5,-751.1,-810.6,-840.8,
     *-840.5,-809.8,-749.9,-662.8,-551.7,-420.6,-274.4,-118.2,42.3,
     *201.2/
      DATA(ZDEM(I,18),I=1,27)/27*0.0/
      DATA(XDEM(I,19),I=1,27)/0.,128.3,254.1,375.1,488.8,593.2,686.3,     R
     *766.2,831.3,880.6,913.,927.9,925.,904.4,866.4,811.9,741.8,657.5,
     *560.5,452.9,336.5,213.7,86.8,-41.7,-169.5,-294.,-412.9/
      DATA(YDEM(I,19),I=1,27)/872.9,864.5,839.6,798.6,742.3,671.7,588.3,
     *493.6,389.4,277.8,160.9,40.8,-80.,-199.2,-314.7,-424.1,-525.4,
     *-616.6,-696.1,-762.1,-813.6,-849.5,-869.1,-872.,-858.2,-828.,
     *-781.9/
      DATA(ZDEM(I,19),I=1,27)/317.7,314.7,305.6,290.7,270.2,244.5,214.1,
     *179.7,141.7,101.1,58.6,14.9,-29.1,-72.5,-114.5,-154.4,-191.2,
     *-224.4,-253.3,-277.4,-296.1,-309.2,-316.3,-317.4,-312.4,-301.4,
     *-284.6/
      DATA(XDEM(I,20),I=1,27)/0.,-96.3,-191.6,-284.9,-375.1,-461.2,       K
     *-542.3,-617.5,-686.3,-747.5,-800.7,-845.2,-880.6,-906.5,-922.7,
     *-928.8,-925.,-911.2,-887.6,-854.3,-811.9,-760.7,-701.3,-634.3,
     *-560.5,-480.7,-395.6/
      DATA(YDEM(I,20),I=1,27)/928.9,923.9,908.9,884.1,849.8,806.3,754.2,
     *693.8,626.,551.5,471.,385.4,295.6,202.7,107.6,11.3,-85.1,-180.6,
     *-274.1,-364.7,-451.3,-533.1,-609.1,-678.6,-740.7,-794.9,-840.4/
      DATA(ZDEM(I,20),I=1,27)/27*0.0/
      END
```

```
      BLOCK DATA TWO
      COMMON/MANEUV/ITT,XDEM(30,45),YDEM(30,45),ZDEM(30,45),
     *              XDTGM(30),YDTGM(30),ZDTGM(30),XPRIME,YPRIME,
     *             ZPRIME,TIMES(50),ITRAJ(60),ITAU(60),III,NADJ
      DATA(XDEM(I,21),I=1,27)/27*0.0/                                    S
      DATA(YDEM(I,21),I=1,27)/928.9,901.7,821.8,693.8,525.3,326.,107.6,
     *-117.1,-334.9,-533.1,-700.2,-826.2,-904.,-928.9,-899.4,-817.3,
     *-687.5,-517.4,-317.,-98.1,126.5,343.8,540.9,706.4,830.6,906.1,
     *928.7/
      DATA(ZDEM(I,21),I=1,27)/0.,-223.,-433.,-617.6,-766.2,-869.8,
     *-922.7,-921.5,-866.4,-760.7,-610.5,-424.5,-213.7,9.6,232.3,441.4,
     *624.7,771.5,873.2,923.7,920.3,863.,755.2,603.2,416.,204.4,-19.1/
      DATA(XDEM(I,22),I=1,27)/0.,160.1,315.3,461.2,593.2,707.5,800.7,   T
     *869.8,913.,928.8,916.9,877.5,811.9,722.,610.5,480.7,336.5,182.3,
     *22.6,-137.7,-294.,-441.4,-575.6,-692.7,-789.,-861.6,-908.6/
      DATA(YDEM(I,22),I=1,27)/928.9,915.,873.8,806.3,714.8,601.9,471.,
     *326.,171.2,11.3,-148.9,-304.7,-451.3,-584.5,-700.2,-794.9,-865.8,
     *-910.8,-928.6,-918.6,-881.2,-817.3,-729.,-618.9,-490.3,-347.,
     *-193.4/
      DATA(ZDEM(I,22),I=1,27)/27*0.0/
      DATA(XDEM(I,23),I=1,27)/0.,-96.3,-191.7,-284.9,-375.1,-461.2,     U
     *-542.3,-617.6,-686.3,-747.5,-800.7,-845.2,-880.6,-906.5,-922.7,
     *-928.8,-925.,-911.2,-887.6,-854.3,-811.9,-760.7,-701.3,-634.3,
     *-560.5,-480.7,-395.6/
      DATA(YDEM(I,23),I=1,27)/804.5,800.1,787.2,765.7,736.,698.3,653.1,
     *600.9,542.2,477.6,407.9,333.8,256.,175.6,93.2,9.8,-73.7,-156.4,
     *-237.4,-315.8,-390.9,-461.7,-527.5,-587.7,-641.5,-688.4,-727.8/
      DATA(ZDEM(I,23),I=1,27)/464.5,462.,454.5,442.1,424.9,403.2,377.1,
     *346.9,313.,275.7,235.5,192.7,147.8,101.4,53.8,5.7,-42.5,-90.3,
     *-137.1,-182.3,-225.7,-266.6,-304.6,-339.3,-370.4,-397.4,-420.2/
      DATA(XDEM(I,24),I=1,27)/0.,-223.,-433.,-617.6,-766.2,-869.8,      V
     *-922.7,-921.5,-866.4,-760.7,-610.5,-424.5,-213.7,9.6,232.3,441.4,
     *624.7,771.5,873.2,923.7,920.2,863.,755.2,603.2,416.,204.4,-19.1/
      DATA(YDEM(I,24),I=1,27)/928.9,901.7,821.8,693.8,525.3,326.,107.6,
     *-117.1,-334.9,-533.1,-700.2,-826.2,-904.,-928.9,-899.4,-817.3,
     *-687.4,-517.4,-317.,-98.1,126.5,343.8,540.9,706.4,830.6,906.1,
     *928.7/
      DATA(ZDEM(I,24),I=1,27)/27*0.0/
      DATA(XDEM(I,25),I=1,27)/0.,-191.6,-375.1,-542.3,-686.3,-800.7,    L
     *-880.6,-922.7,-925.,-887.6,-811.9,-701.3,-560.5,-395.6,-213.7,
     *-22.6,169.5,354.3,523.8,670.8,789.,873.2,919.8,926.8,894.,822.6,
     *715.9/
      DATA(YDEM(I,25),I=1,27)/928.9,908.9,849.8,754.2,626.,471.,295.6,
     *107.6,-85.1,-274.1,-451.3,-609.1,-740.7,-840.4,-904.,-928.6,
     *-913.3,-858.7,-767.1,-642.6,-490.3,-315.,-130.,62.5,252.4,431.4,
     *591.9/
      DATA(ZDEM(I,25),I=1,27)/27*0.0/
      DATA(XDEM(I,26),I=1,27)/0.,223.,433.,617.6,766.2,869.8,922.7,     M
     *921.5,866.4,760.7,610.5,424.5,213.7,-9.6,-232.3,-441.4,-624.7,
     *-771.5,-873.2,-923.7,-920.2,-863.,-755.2,-603.2,-416.,-204.4,19.1/
      DATA(YDEM(I,26),I=1,27)/928.9,901.7,821.8,693.8,525.3,326.,107.6,
     *-117.1,-334.9,-533.1,-700.2,-826.2,-904.,-928.9,-899.4,-817.3,
     *-687.4,-517.4,-317.,-98.1,126.5,343.8,540.9,706.4,830.6,906.1,
     *928.7/
      DATA(ZDEM(I,26),I=1,27)/27*0.0/
      DATA(XDEM(I,27),I=1,27)/0.,191.6,375.1,542.3,686.3,800.7,880.6,   N
     *922.7,925.,887.6,811.9,701.3,560.5,395.6,213.7,22.6,-169.5,-354.3,
```

```
     *-523.8,-670.8,-789.,-873.2,-919.8,-926.8,-894.,-822.6,-715.9/
      DATA(YDEM(I,27),I=1,27)/928.9,908.9,849.8,754.2,626.,471.,295.6,
     *107.6,-85.1,-274.1,-451.3,-609.1,-740.7,-840.4,-904.,-928.6,
     *-913.3,-858.7,-767.1,-642.6,-490.3,-317.,-130.,62.5,252.4,
     *431.4,591.9/
      DATA(ZDEM(I,27),I=1,27)/27*0.0/
      DATA(XDEM(I,28),I=1,27)/27*0.0/                                    PP
      DATA(YDEM(I,28),I=1,27)/27*928.9/
      DATA(ZDEM(I,28),I=1,27)/27*0.0/
      DATA(XDEM(I,29),I=1,27)/27*0.0/                                    W
      DATA(YDEM(I,29),I=1,27)/928.9,908.9,849.8,754.2,626.,471.,295.6,
     *107.6,-85.1,-274.1,-451.3,-609.1,-740.7,-840.4,-904.,-928.6,
     *-913.3,-858.7,-767.1,-642.6,-490.3,-317.,-130.,62.5,252.4,431.4,
     *591.9/
      DATA(ZDEM(I,29),I=1,27)/0.,-191.7,-375.1,-542.3,-686.3,-800.7,
     *-880.6,-922.7,-925.,-887.6,-811.9,-701.3,-560.5,-395.6,-213.7,
     *-22.6,169.5,354.3,523.8,670.8,789.,873.2,919.8,925.8,894.,822.6,
     *715.9/
      DATA(XDEM(I,30),I=1,27)/0.,96.3,191.7,284.9,375.1,461.2,542.3,     O
     *617.6,686.3,747.5,800.7,845.2,880.6,906.5,922.7,928.8,925.,911.2,
     *887.6,854.3,811.9,760.7,701.3,634.3,560.5,480.7,395.6/
      DATA(YDEM(I,30),I=1,27)/804.5,800.1,787.2,765.7,736.,698.3,653.1,
     *600.9,542.2,477.6,407.9,333.8,256.,175.6,93.2,9.8,-73.7,-156.4,
     *-237.4,-315.8,-390.9,-461.7,-527.5,-587.7,-641.5,-688.4,-727.8/
      DATA(ZDEM(I,30),I=1,27)/464.5,462.,454.5,442.1,424.9,403.2,377.1,
     *346.9,313.,275.7,235.5,192.7,147.8,101.4,53.8,5.7,-42.5,-90.3,
     *-137.1,-182.3,-225.7,-266.6,-304.6,-339.3,-370.4,-397.4,-420.2/
      DATA(XDEM(I,31),I=1,27)/0.,-160.1,-315.3,-461.2,-593.2,-707.5,     X
     *-800.7,-869.8,-913.,-928.8,-916.9,-877.5,-811.9,-722.,-610.5,
     *-480.7,-336.5,-182.3,-22.6,137.7,294.,441.4,575.6,692.7,789.,
     *861.6,908.6/
      DATA(YDEM(I,31),I=1,27)/928.9,915.,873.8,806.3,714.8,601.9,471.,
     *326.,171.2,11.3,-148.9,-304.7,-451.3,-584.5,-700.2,-794.9,-865.8,
     *-910.8,-928.6,-918.5,-881.2,-817.3,-729.,-618.9,-490.3,-347.,
     *-193.4/
      DATA(ZDEM(I,31),I=1,27)/27*0.0/
      DATA(XDEM(I,32),I=1,27)/0.,-128.3,-254.1,-375.1,-488.8,-593.2,     Y
     *-686.3,-766.2,-831.3,-880.6,-913.,-927.9,-925.,-904.4,-866.4,
     *-811.9,-741.8,-657.5,-550.5,-452.9,-336.5,-213.7,-86.8,41.7,169.5,
     *294.,412.9/
      DATA(YDEM(I,32),I=1,27)/872.9,864.5,839.6,798.6,742.3,671.7,588.3,
     *493.6,389.4,277.8,160.9,40.8,-80.,-199.2,-314.7,-424.1,-525.4,
     *616.6,-696.1,-762.1,-813.6,-849.5,-869.1,-872.,-858.2,828.,-781.9/
      DATA(ZDEM(I,32),I=1,27)/317.7,314.7,305.6,290.7,270.2,244.5,214.1,
     *179.7,141.7,101.1,58.6,14.9,-29.1,-72.5,-114.5,-154.4,-191.2,
     *-224.4,-253.3,-277.4,-296.1,-309.2,-316.3,-317.4,-312.4,-301.4,
     *-284.6/
      DATA(XDEM(I,33),I=1,27)/27*0.0/                                    P
      DATA(YDEM(I,33),I=1,27)/1013.4,1011.3,1005.2,995.,980.9,962.7,
     *940.7,914.9,885.4,852.3,815.8,776.,733.1,687.3,638.6,587.4,533.8,
     *478.1,420.4,361.1,300.3,239.2,175.2,111.6,47.4,-16.9,-81.2/
      DATA(ZDEM(I,33),I=1,27)/0.,-64.3,-128.4,-191.9,-254.9,-316.4,
     *-376.8,-435.7,-492.9,-548.1,-601.1,-651.7,-699.6,-744.7,-786.8,
     *-825.8,-861.4,-893.5,-922.,-946.9,-967.9,-985.,-998.1,-1007.2,
     *-1012.3,-1013.2,-1010.1/
      DATA(XDEM(I,34),I=1,27)/27*0.0/                                    QQ
      DATA(YDEM(I,34),I=1,27)/27*978.8/
```

```
      DATA(ZDEM(I,34),I=1,27)/27*262.3/
      DATA(XDEM(I,35),I=1,27)/27*0.0/                                    Z
      DATA(YDEM(I,35),I=1,27)/1013.4,988.4,914.9,796.3,638.6,449.5,
     *238.2,15.3,-208.5,-421.9,-614.6,-777.1,-901.4,-981.3,-1012.9,
     *-994.7,-927.6,-814.9,-652.,-476.6,-267.8,-45.8,178.5,394.,590.1,
     *757.2,887./
      DATA(ZDEM(I,35),I=1,27)/0.,-223.4,-435.7,-626.7,-786.8,-908.2,
     *-985.,-1013.2,-991.7,-921.4,-805.7,-650.4,-463.1,-253.,-30.5,
     *193.5,408.,602.4,767.2,894.3,977.3,1012.3,997.5,933.6,823.8,
     *673.5,490./
      DATA(XDEM(I,36),I=1,27)/27*0.0/                                    AA
      DATA(YDEM(I,36),I=1,27)/1013.4,1008.8,995.,972.3,940.7,900.6,
     *852.3,796.3,733.1,663.3,587.4,506.2,420.4,330.8,238.2,143.5,
     *47.4,-49.1,-145.1,-239.8,-332.4,-421.9,-507.6,-588.7,-664.5,
     *-734.3,-797.4/
      DATA(ZDEM(I,36),I=1,27)/0.,-96.4,-191.9,-285.6,-376.8,-464.6,
     *-548.1,-626.7,-699.6,-766.1,-825.8,-877.9,-922.,-957.8,-985.,
     *-1003.2,-1012.3,-1012.2,-1002.9,-984.6,-957.3,-921.4,-877.,-824.8,
     *-765.1,-698.4,-625.4/
      DATA(XDEM(I,37),I=1,27)/0.,128.4,254.6,376.8,492.9,601.1,699.6,    BB
     *786.8,861.4,922.,967.9,998.1,1012.2,1010.1,991.7,957.3,907.5,
     *843.1,765.1,674.7,573.5,463.1,345.2,221.8,94.7,-33.8,-161.8/
      DATA(YDEM(I,37),I=1,27)/1013.4,1005.2,980.8,940.7,885.4,815.8,
     *733.1,638.6,533.8,420.4,300.2,175.2,47.4,-81.2,-208.4,-332.4,
     *-451.,-562.3,-664.5,-756.1,-835.4,-901.4,-952.8,-988.8,-1008.9,
     *-1012.8,-1000.4/
      DATA(ZDEM(I,37),I=1,27)/27*0.0/
      DATA(XDEM(I,38),I=1,27)/27*0.0/                                    CC
      DATA(YDEM(I,38),I=1,27)/1013.4,995.,940.7,852.3,733.1,587.4,420.4,
     *238.2,47.4,-145.1,-332.4,-507.6,-664.5,-797.4,-901.4,-972.7,
     *-1008.9,-1008.6,-971.8,-899.8,-795.3,-662.,-504.8,-329.3,-141.8,
     *50.7,241.4/
      DATA(ZDEM(I,38),I=1,27)/0.,-191.9,-376.8,-548.1,-699.6,-825.8,
     *-922.,-985.,-1012.3,-1002.9,-957.3,-877.,-765.1,-625.4,-463.1,
     *-284.1,-94.7,98.,287.2,466.,628.,767.2,878.7,958.4,1003.4,1012.1,
     *984.2/
      DATA(XDEM(I,39),I=1,27)/27*0.0/                                    RR
      DATA(YDEM(I,39),I=1,27)/27*1013.4/
      DATA(ZDEM(I,39),I=1,27)/27*0.0/
      DATA(XDEM(I,40),I=1,27)/0.,-64.3,-128.4,-191.9,-254.6,-316.4,      DD
     *-376.8,-435.7,-492.9,-548.1,-601.1,-651.7,-699.6,-744.7,-786.8,
     *-825.8,-861.4,-893.5,-922.,-946.9,-967.9,-985.,-998.1,-1007.2,
     *-1012.3,-1013.2,-1010.1/
      DATA(YDEM(I,40),I=1,27)/877.6,875.8,870.5,861.7,849.4,833.7,814.7,
     *792.3,766.8,738.1,706.5,672.1,643.9,595.2,553.1,508.7,462.3,414.,
     *364.1,312.7,260.,206.3,151.8,96.6,41.1,-14.6,-70.3/
      DATA(ZDEM(I,40),I=1,27)/506.7,505.7,502.6,497.5,490.4,481.4,470.4,
     *457.4,442.7,426.2,407.9,388.,366.6,343.6,319.3,293.7,266.9,239.,
     *210.2,180.5,150.1,119.1,87.6,55.8,23.7,-8.5,-40.6/
      DATA(XDEM(I,41),I=1,27)/0.,64.3,128.4,191.9,254.6,316.4,376.8,     EE
     *435.7,492.9,548.1,601.1,651.7,699.6,744.7,786.8,825.8,861.4,893.5,
     *922.,946.9,967.9,985.,998.1,1007.2,1012.3,1013.2,1010.1/
      DATA(YDEM(I,41),I=1,27)/877.6,875.8,870.5,861.7,849.4,833.7,814.7,
     *792.3,766.8,738.1,706.5,672.1,634.9,595.2,553.1,508.7,462.3,414.,
     *364.1,312.7,260.,206.3,151.8,96.6,41.1,-14.6,-70.3/
      DATA(ZDEM(I,41),I=1,27)/506.7,505.7,502.6,497.5,490.4,481.4,470.4,
     *457.4,442.7,426.2,407.9,388.,366.6,343.6,319.3,293.7,266.9,239.,
```

```
DATA TWO          74/74    OPT=1                    FTNH 4.2+75075        07/16/75

      *210.2,180.5,150.1,119.1,87.6,55.8,23.7,-8.5,-40.6/
      END
```

```
      BLOCK DATA THREE
      COMMON/COMA/LEVEL,IPTS,XXS(50),XDTGO,YDTGO,ZDTGO,RLB,COSE,SPO,RI,
     *GAM,EDOT,THETAL,RN,
     *              PPX(50),PPY(50),PPZ(50),TIME(50),TMAS(30),XDTGMS(30)
     *             ,YDTGMS(30),ZDTGMS(30),XMAN(4,50),XMISS(7),NT
     *             ,XCOMP,YCOMP,ZCOMP,TAMA(30),DELTAR(30),VM(30),G,GGG
     *,XDO,YDO,ZDO,DXG,DYG,DZG,S2,S3,S4,S5,XDM(30),YDM(30),ZDM(30),
     *RLBK,SCALEP,F1,F2,F3,G1,G2,G3,XC,YC,ZC,S1,RRR,SR,SPL,CTL,STL,
     *CPL,A1,VTI,XE(30),YE(30),ZE(30),ZALT,NERR,CLA,NPX,NX,CLAA(10),
     *PHO,ARG,AAA,SCALET,TREAL,TMA(30),XTA,YTA,ZTA,SCALEV,QMM(10),QM
     *,SA,CA,VMX(50),VMY(50),VMZ(50)
     *,ECOS(17),HR7EE(17),FI7T(9),FCI77(9)
      COMMON/PK/XDH(22),YDH(22),ZDH(22),BDHX(22),BDHY(22),BDHZ(22)
     *,KILL(22),P(4),SPK,SPHG,SPHS,SPM
      COMMON/MANEUV/ITT,XDEM(30,45),YDEM(30,45),ZDEM(30,45),
     *              XDTGM(30),YDTGM(30),ZDTGM(30),XPRIME,YPRIME,
     *             ZPRIME,TIMES(50),ITRAJ(60),ITAU(60),III,NADJ
      DATA(XDEM(I,42),I=1,27)/0.,-160.2,-316.4,-464.6,-601.1,-722.5,      FF
     *-825.8,-908.2,-967.9,-1003.2,-1013.2,-997.8,-957.3,-892.7,-805.7,
     *-698.4,-573.5,-434.2,-284.0,-126.7,33.8,193.5,348.3,494.4,628.,
     *745.8,844.9/
      DATA(YDEM(I,42),I=1,27)/1013.4,1000.6,962.7,900.6,815.8,710.6,
     *587.4,449.5,300.2,143.5,-16.9,-176.9,-332.4,-479.5,-614.6,-734.3,
     *-835.4,-915.6,972.7,-1005.4,-1012.8,-994.7,-951.6,-884.6,-795.3,
     *-686.,-559.5/
      DATA(ZDEM(I,42),I=1,27)/27*0.0/
      DATA(XDEM(I,43),I=1,27)/0.,191.9,376.8,548.1,699.6,825.8,922.,985.  GG
     *,1012.3,1002.9,957.3,877.,765.1,625.4,463.1,284.1,94.7,-98.,-287.2
     *,-466.,-628.,-767.2,-878.7,-958.4,-1003.4,-1012.1,-984.2/
      DATA(YDEM(I,43),I=1,27)/978.8,961.1,908.6,823.3,708.1,567.4,406.1,
     *230.1,45.8,-140.2,-321.1,-490.3,-641.9,-770.2,-870.6,-939.6,-974.5
     *,-974.2,-938.7,-869.2,-768.2,-639.5,-487.6,-318.,-137.,49.,233.2/
      DATA(ZDEM(I,43),I=1,27)/262.3,257.5,243.5,220.6,189.7,152.,108.8,
     *61.7,12.3,-37.6,-86.,-131.4,-172.,-206.4,-233.3,-251.8,-261.1,
     *-261.,-251.5,-232.9,-205.8,-171.3,-130.6,-85.2,-36.7,13.1,62.5/
      DATA(XDEM(I,44),I=1,27)/27*0.0/                                      HH
      DATA(YDEM(I,44),I=1,27)/952.3,1005.2,1008.8,962.7,869.2,733.,
     *560.7,360.9,143.3,-81.4,-302.,-507.8,-688.6,-835.5,-941.4,-1000.9,
     *-1011.2,-971.8,-884.5,-753.7,-585.9,-389.2,-173.4,50.9,272.7,
     *481.2,665.9/
      DATA(ZDEM(I,44),I=1,27)/346.6,128.2,-96.6,-316.5,-520.9,-699.7,
     *-844.1,-946.9,-1003.2,-1010.1,-967.3,-877.,-743.5,-573.4,-375.1,
     *-158.4,66.1,287.4,494.3,677.3,826.8,935.6,998.4,1012.1,976.,891.9,
     *763.9/
      DATA(XDEM(I,45),I=1,27)/27*0.0/                                      SS
      DATA(YDEM(I,45),I=1,27)/27*1124.8/
      DATA(ZDEM(I,45),I=1,27)/27*0.0/
      DATA  (CLAA(I),I=1,8)/0.043,0.0435,0.0455,0.0485,0.0515,0.059,
     *      0.0477,0.033/
      DATA  (QMM(I),I=1,8)/0.0,0.3,0.6,0.8,0.9,1.05,1.5,2.2/
      DATA(TAMA(I),I=1,28)/0.,.25,1.,1.7,2.,2.5,3.,4.,4.5,5.,6.,7.,
     *    7.5,8.,9.,9.5,10.,11.,11.5,12.,13.,14.,15.,16.,17.,18.,19.,
     *    20./
      DATA(DELTAR(I),I=1,28)/0.,31.2,445.6,1418.6,1978.1,2958.1,3945.6,
     *    5870.6,6805.6,7728.1,9558.1,11368.1,12263.,13115.,14580.,
     *    15188.,15738.,15753.,17218.,17660.,18505.,19295.,20038.,
     *    20745.,21420.,22060.,22665.,23240./
```

```
      DATA(VM(I),I=1,28)/125.,125.,980.,1800.,1930.,1990.,1960.,1890.,
     *    1850.,1840.,1820.,1800.,1780.,1630.,1300.,1130.,1070.,960.,
     *    900.,870.,820.,760.,725.,690.,660.,620.,590.,560./
      DATA (XOH(I),I=1,22)/13.5,0.,9.,-11.3,-16.8,-13.2,-1.5,-4.2,-16.3,
     *-18.2,-20.3,-6.8,-19.4,-19.4,-19.8,3.4,-1.6,-5.6,-1.6,-16.7,-12.2,
     *-16.3/
      DATA (YOH(I),I=1,22)/12*0.,-4.,4.,0.,0.,.9,0.,-.9,3*0./
      DATA (ZOH(I),I=1,22)/.1,.2,.6,0.,3.8,2.8,-.3,-.3,.3,.4,5.7,-.3,.3,
     *.3,0.,-1.6,-1.6,-1.6,-1.6,-.2,-1.9,-.1/
      DATA (BOHX(I),I=1,22)/6.5,12.5,6.,6.5,2.7,2.8,6.8,3.,1.8,1.5,2.5,
     *2.6,2.2,2.2,.2,1.3,1.3,1.3,1.3,3.1,3.6,2./
      DATA (BOHY(I),I=1,22)/1.7,2.,1.9,2.,.3,.4,7.5,9.5,3.2,.3,.1,11.9,
     *2.2,2.2,1.,1.6,.6,1.6,.6,1.9,.3,2.1/
      DATA (BOHZ(I),I=1,22)/2.,2.5,2.8,2.3,2.7,1.6,.3,.3,.2,2.3,2.3,.2,
     *.1,.1,1.,.9,.6,.9,.6,1.9,1.4,2.1/
      DATA (KILL(I),I=1,22)/4*1,0,1,0,0,1,5*0,1,5*0,1,1/
      DATA (ECOS(I),I=1,17)/-1.,-.9848,-.9397,-.866,-.7071,-.5,-.342,
     *-.1736,0.0,.1736,.5,.7071,.866,.9397,.9848,.9962,1./
      DATA (HR7EE(I),I=1,17)/5.3568E-10,6.4385E-10,5.7864E-10,3.7351E-10
     *,1.7936E-10,6.8327E-11,4.0449E-11,2.8362E-11,2.0376E-11,2.027E-11,
     *1.8443E-11,1.568E-11,1.184E-11,8.7428E-12,5.2709E-12,3.4869E-12,1.
     *8586E-12/
      DATA(FI7T(I),I=1,9)/.18,.332,.612,.737,.833,.92,.97,.99999,1.0000/
      DATA (FCI77(I),I=1,9)/0.0,.10491819,.31475458,.41967277,.52459096,
     *.62950916,.73442735,.78689133,.97680098/
      END
```

```
      SUBROUTINE INIT
      COMMON/EXTRA/NPTS,DX,XTERM,KKK,AS,SEO,SC,ASEO,R,SCALEL,VVVV,C1,
     *XLOS,YLOS,ZLOS,CO,CX,CY,CLY,CLZ,TRP,XX(50),THE
      COMMON/COMA/LEVEL,IPTS,XXS(50),XDTGO,YDTGO,ZDTGO,RLB,COSE,SPO,RI,
     *GAM,EDOT,THETAL,RN,
     *              PPX(50),PPY(50),PPZ(50),TIME(50),TMAS(30),XDTGMS(30)
     *             ,YDTGMS(30),ZDTGMS(30),XMAN(4,50),XMISS(7),NT
     *             ,XCOMP,YCOMP,ZCOMP,TAMA(30),DELTAR(30),VM(30),G,GGG
     *,XDO,YDO,ZDO,DXG,DYG,DZG,S2,S3,S4,S5,XDM(30),YDM(30),ZDM(30),
     *RLBK,SCALEP,F1,F2,F3,G1,G2,G3,XC,YC,ZC,S1,RRR,SR,SPL,CTL,STL,
     *CPL,A1,VTI,XE(30),YE(30),ZE(30),ZALT,NERR,CLA,NPX,NX,CLAA(10),
     *PHO,ARG,AAA,SCALET,TREAL,TMA(30),XTA,YTA,ZTA,SCALEV,QMM(10),QM
     *,SA,CA,VMX(50),VMY(50),VMZ(50)
     *,ECOS(17),HR7EE(17),FI7T(9),FCI77(9)
      COMMON /COMB/XD,YD,ZD,XDTG,YDTG,ZDTG,XDEF,YDEF,ZDEF,ZZZ
     *,XDDE,YDDE,ZDDE
      COMMON/COMP/IS1,I99,I88,CLMS(10),CLS1(10),CLS2(10),NSAM,Q1,Q2,Q3,
     *Q4,Q5,Q6,W1,W2,W3,W4,W5,W6,IFT,IPLANE,ICON,S2P1,S1P1,A2P1,A1P1,
     *S2P2,S1P2,A2P2,A1P2,S2P3,S1P3,A2P3,A1P3,S2P4,S1P4,A2P4,A1P4,S2P5,
     *S1P5,A2P5,A1P5,S2P6,S1P6,A2P6,A1P6
      COMMON/MANEUV/ITT,XDEM(30,45),YDEM(30,45),ZDEM(30,45),
     *              XDTGM(30),YDTGM(30),ZDTGM(30),XPRIME,YPRIME,
     *             ZPRIME,TIMES(50),ITRAJ(60),ITAU(60),III,NADJ
      COMMON/GTARG/FXA,FYA,FZA,FXB,FYB,FZB,FXC,FYC,FZC,AT,VT
      DATA  RTD,DTR/57.2957795,0.01745329/
C*** GENERATE THE DISTANCE TABLE FOR COLLECTING DATA POINTS
      SCALEP=20.*(102.375)/(1.+GGG)
      STPN=NPTS
      DIF=(XTERM+(STPN-1.)*DX)-((20.*102.375)/(1.+GGG))
      IF(DIF.GE.0.) GO TO 47
      NADJ=NPTS
      GO TO 48
   47 ARG=DIF/DX
      NADJ=NPTS-IFIX(ARG)-1
   48 CONTINUE
      DO 20 I=1,NADJ
      INDEX=NADJ-I+1
      XI = I - 1
      XX(INDEX) = XTERM + (XI*DX)
      XXS(INDEX)=-XX(INDEX)/SCALEP
   20 CONTINUE
C*** WRITE OUT RANGE TABLE
      IF(IS1.NE.1) PRINT 820, NPTS,DX,XTERM
      IF(IS1.EQ.1) GO TO 21
      PRINT 800
      PRINT 700
      PRINT 800
      PRINT 705
      PRINT 800
      PRINT 822, (XX(I),XXS(I),I=1,NADJ)
      PRINT 800
   21 CONTINUE
      XD=XDM(1)
      YD=YDM(1)
      ZD=ZDM(1)
      XDX=XD
      YDY=YD
```

```
      ZDZ=ZD
      CALL GUID
      XDTGO=XDTG
      YDTGO=YDTG
      ZDTGO=ZDTG
      PRINT 806,XD,YD,ZD,XDTG,YDTG,ZDTG
      ARG = SQRT (XD**2 + YD**2)
      IF(KKK.EQ.1) GO TO 65
      SEO = ATAN2 (ZD,ARG)
   65 SC=AS*DTR
      ASEO = SEO * RTD
      IF(ITT.NE.1) PRINT 807,AS,ASEO
C*** SCALE INITIAL LAUNCH CONDITIONS
      RFEET = RI
      RI=R/SCALEL
      GAM=G/SCALET
      XDTGO=XDTGO/SCALEV
      YDTGO=YDTGO/SCALEV
      ZDTGO=ZDTGO/SCALEV
C*** TRANSFORM MANEUVER VECTORS TO LAUNCH COORDINATES AND SCALE THEM
      IF(IS1.EQ.1) GO TO 66
      PRINT 800
      PRINT 910
      PRINT 800
      PRINT 907
   66 CONTINUE
      DO 70 I = 1,NT
      XD=XDM(I)
      YD=YDM(I)
      ZD=ZDM(I)
      CALL GUID
      XDTGM(I) = XDTG
      YDTGM(I) = YDTG
      ZDTGM(I) = ZDTG
      XDTGMS(I) = XDTGM(I) / SCALEV
      YDTGMS(I) = YDTGM(I) / SCALEV
      ZDTGMS(I) = ZDTGM(I) / SCALEV
      IF(IS1.NE.1) PRINT 908, TMA(I),XDM(I),YDM(I),ZDM(I)
   70 CONTINUE
      XD=XDX
      YD=YDY
      ZD=ZDZ
      IF(IS1.EQ.1) GO TO 76
      PRINT 911
      PRINT 800
      PRINT 907
      PRINT 800
      DO 75 I=1,NT
      PRINT 908, TMA(I),XDTGM(I),YDTGM(I),ZDTGM(I)
   75 CONTINUE
   76 CONTINUE
C     CALCULATE INITIAL TARGET ANGLE OF ATTACK
      AAA=1116.89+0.003894*ZZZ
      ARG=0.00003*ZZZ
      PHO=0.00237692*EXP(ARG)
      QM=VVVV/AAA
      NPX=2
```

```
      NX=8
      CALL INTERP (QM,QMM,CLAA,NX,NPX,CLA,NERR)
      WT=18500.0
      ST=248.0
      XEU=XE(1)
      YEU=YE(1)
      ZEU=ZE(1)
      AT=2.*WT*SQRT(XEU**2+YEU**2+ZEU**2)/(PHO*(VVVV**2)*ST*CLA*32.174)
      AT=AT*DTR
C     CALCULATE INITIAL COSE
      C1=COS(AT)
      S1=SIN(AT)
      S9=SQRT(XD*XD+YD*YD)
      COSE=(STL*(ZD*C1-S9*S1)-(C1+(ZD*S1/S9))*(XD*S2+YD*S4))/VVVV
      E111 = COSE
      CO=COSE**2
      COSE=COSE/1.02375
C     CALCULATE INITIAL APPARENT PLUME LENGTH-TO-BREADTH RATIO
      RLB=RLBK*SQRT(1.-CO)
      RLB=RLB/SCALET
C     CALCULATE INITIAL LOS
      XLOS=-R*S2
      YLOS=-R*S4
      ZLOS=R*STL
C     CALCULATE HORIZON REFERENCE VECTOR
      ARG=YLOS/SQRT(XLOS**2+YLOS**2)
      BET=ACOS(ARG)
      IF(XLOS.LT.0.0) CY=SIN(BET)
      IF(XLOS.GE.0.0) CY=-SIN(BET)
      CX=-COS(BET)
C     CALCULATE INITIAL TARGET CENTERLINE UNIT VECTOR
      CLY=YD*C1/VVVV
      CLZ=-SQRT(YD**2)*S1/VVVV
C     CALCULATE INITIAL PLUME ROTATION ANGLE
      ARG1=(YLOS*CLZ-ZLOS*CLY)*CX+(ZLOS*CLX-XLOS*CLZ)*CY
      ARG2=SQRT((YLOS*CLZ-ZLOS*CLY)**2+(ZLOS*CLX-XLOS*CLZ)**2
     *+(XLOS*CLY-YLOS*CLX)**2)
      ARG=ARG1/ARG2
      THE=ACOS(ARG)
      ARG=XLOS*CLY-YLOS*CLX
      IF(ARG.LT.0.0) GO TO 80
      IF(THE.LE.1.570796326) GO TO 85
      TRP=THE-4.71238898
      GO TO 90
   80 TRP=1.570796326-THE
      GO TO 90
   85 TRP=1.570796326+THE
   90 SPO=TRP/SCALET
      NX=17
      NPX=2
      CALL INTERP(E111,ECOS,HR7EE,NX,NPX,HR7E,NERR)
      HC7FT=1209.675*HR7E*EXP(-2.341*ALOG(RFEET/22965.831))
      P1IRSS=(2.0*ATAN(0.04884004884/RLB))
      P2IRSS=230.34375*RLB/RFEET
      AT7   =((P2IRSS**2)*SIN(P1IRSS))/(2.0*(COS(P1IRSS/2.0))**2)
      PJTU1=0.1*EXP(1.003258*ALOG(AT7))
      FI7=HC7FT/PJTU1
```

```
      NX=9
      NPX=2
      CALL INTERP (FI7,FI7T,FCI77,NX,NPX,FCI7,NERR)
      RN=FCI7
  700 FORMAT(10X,38H RANGE-TO-GO TABLE FOR DATA COLLECTION)
  705 FORMAT(22X,9H RANGE,FT,5X,13H SCALED RANGE)
  800 FORMAT (1H0)
  806 FORMAT(10X,8H XDTEO =F10.2,7H FT/SEC,5X,8H YDTEO =F10.2,7H FT/SEC,
     *5X,8H ZDTEO =F10.2,7H FT/SEC,/10X,8H XDTLO =F10.2,7H FT/SEC,5X,
     *8H YDTLO =F10.2,7H FT/SEC,5X,8H ZDTLO =F10.2,7H FT/SEC)
  807 FORMAT(10X,17H CROSSING ANGLE =F7.2,4H DEG,5X,13H DIVE ANGLE =
     *F7.2,4H DEG)
  820 FORMAT(10X,7H NPTS =I3,4X,5H DX =F6.2,3H FT,4X,8H XTERM =F7.2,
     *3H FT)
  822 FORMAT (20X F10.2,5X E15.8)
  907 FORMAT (1H0 4X 35H TIME    XD MANEUVER    YD MANEUVER
     *                  15H    ZD MANEUVER)
  908 FORMAT(1X F10.4, 3(3X F8.2,4X))
  910 FORMAT(10X,24H EARTH FIXED COORDINATES)
  911 FORMAT(10X,19H LAUNCH COORDINATES)
      RETURN
      END
```

```
      SUBROUTINE LETH
      COMMON/PK/XDH(22),YDH(22),ZDH(22),BDHX(22),BDHY(22),BDHZ(22)
     *,KILL(22),P(4),SPK,SPHG,SPHS,SPM
      COMMON/MISD/XM(3),YM(3),ZM(3),RM(3),NOPT,ICN
      COMMON/MANEUV/ITT,XDEM(30,45),YDEM(30,45),ZDEM(30,45),
     *              XDTGM(30),YDTGM(30),ZDTGM(30),XPRIME,YPRIME,
     *             ZPRIME,TIMES(50),ITRAJ(60),ITAU(60),III,NADJ
      COMMON/DELT/PPXS(50),PPYS(50),PPZS(50),VMXS(50),VMYS(50),VMZS(50)
      COMMON/STARG/FXA,FYA,FZA,FXB,FYB,FZB,FXC,FYC,FZC,AT,VT
      COMMON/COMD/COSA,COSB,COSG,YA,ZA
      COMMON/COMP/IS1,I99,I88,CLMS(10),CLS1(10),CLS2(10),NSAM,Q1,Q2,Q3,
     *Q4,Q5,Q6,W1,W2,W3,W4,W5,W6,IFT,IPLANE,ICON,S2P1,S1P1,A2P1,A1P1,
     *S2P2,S1P2,A2P2,A1P2,S2P3,S1P3,A2P3,A1P3,S2P4,S1P4,A2P4,A1P4,S2P5,
     *S1P5,A2P5,A1P5,S2P6,S1P6,A2P6,A1P6
      COSA=(1./ZA)/SQRT((1./ZA)**2+(YA/ZA)**2+1.)
      COSB=(YA/ZA)/SQRT((1./ZA)**2+(YA/ZA)**2+1.)
      COSG=1./SQRT((1./ZA)**2+(YA/ZA)**2+1.)
      SSX=0.0
      SSY=0.0
      SSZ=0.0
      DO 4 J=1,NADJ
      SSX=SSX+VMXS(J)
      SSY=SSY+VMYS(J)
      SSZ=SSZ+VMZS(J)
    4 CONTINUE
      ADJ=FLOAT(NADJ)
C***  AVERAGE LAUNCH COORDINATES OF RELATIVE VELOCITY VECTORS OVER DATA
C     COLLECTION RANGE
      COOL5=(SSX/ADJ)
      COOL6=(SSY/ADJ)
      COOL7=(SSZ/ADJ)
C***  GENERALIZED TARGET COORDINATES OF AVERAGE RELATIVE VELOCITY VECTOR
C     REFERENCED TO ORIGIN AT TAILPIPE
      CUUL5=FXA*COOL5+FYA*COOL6+FZA*COOL7
      CUUL6=FXB*COOL5+FYB*COOL6+FZB*COOL7
      CUUL7=FXC*COOL5+FYC*COOL6+FZC*COOL7
      C1=COS(AT)
      S1=SIN(AT)
C***  TARGET FIXED COORDINATES OF AVERAGE RELATIVE VELOCITY VECTOR
C     REFERENCED TO ORIGIN AT TAILPIPE
      CARL5=CUUL7*C1+CUUL5*S1
      CARL6=-CUUL6
      CARL7=CUUL5*C1-CUUL7*S1
C***  TARGET FIXED COORDINATES OF AVERAGE INERTIAL MISSILE VELOCITY
C     VECTOR
C     REFERENCED TO ORIGIN AT TAILPIPE
      VM1=(CUUL7+VT)*C1+CUUL5*S1
      VM2=-CUUL6
      VM3=CUUL5*C1-(CUUL7+VT)*S1
C***  GENERALIZED TARGET COORDINATES OF POSITION VECTOR OF LAST DATA
C     POINT
C     REFERENCED TO ORIGIN AT TAILPIPE
      XAI=FXA*PPXS(NADJ)+FYA*PPYS(NADJ)+FZA*PPZS(NADJ)
      ZAI=FXC*PPXS(NADJ)+FYC*PPYS(NADJ)+FZC*PPZS(NADJ)
      YAI=FXB*PPXS(NADJ)+FYB*PPYS(NADJ)+FZB*PPZS(NADJ)
C***  TARGET FIXED COORDINATES OF POSITION VECTOR OF LAST DATA POINT
C     REFERENCED TO ORIGIN AT TAILPIPE
```

```
      XAIMD=ZAI*C1+XAI*S1
      YAIMD=-YAI
      ZAIMD=XAI*C1-ZAI*S1
C***  DISTANCE BETWEEN LAST DATA POINT AND XF-YF PLANE PIERCING POINT
C     REFERENCED TO ORIGIN AT TAILPIPE
      D1=SQRT((XAIMD+ZM(1)/ZA)**2+(YAIMD-YM(1)+(YA/ZA)*ZM(1))**2
     **+ZAIMD**2)
C***  COMPONENTS OF UNIT VECTOR IN DIRECTION FROM XF-YF PLANE PIERCING
C     POINT TO POSITION OF LAST DATA POINT
C     REFERENCED TO ORIGIN AT TAILPIPE
      AA=(XAIMD+ZM(1)/ZA)/D1
      BB=(YAIMD-YM(1)+(YA/ZA)*ZM(1))/D1
      CC=ZAIMD/D1
      IF(COSA/ABS(COSA).NE.AA/ABS(AA)) COSA=-COSA
      IF(COSB/ABS(COSB).NE.BB/ABS(BB)) COSB=-COSB
      IF(COSG/ABS(COSG).NE.CC/ABS(CC)) COSG=-COSG
C***  COSINE OF THETA
      COST=AA*COSA+BB*COSB+CC*COSG
C***  DISTANCE BETWEEN PK MODEL AIM POINT AND XF-YF PLANE PIERCING POINT
C     REFERENCED TO ORIGIN AT TAILPIPE
      D2=D1*COST
C***  TARGET FIXED COORDINATES OF PK MODEL AIM POINT
C     REFERENCED TO ORIGIN AT CG
      XAIM=D2*COSA-ZM(1)/ZA-19.5
      YAIM=D2*COSB+YM(1)-(YA/ZA)*ZM(1)
      ZAIM=D2*COSG
      IF(IS1.EQ.1) GO TO 300
      PRINT 999,XAIMD,YAIMD,ZAIMD,D1,AA,BB,CC,COST,D2
      PRINT 999,COSA,COSB,COSG
  300 CONTINUE
      NELIPS=22
      VSHELL=SQRT(VM1**2+VM2**2+VM3**2)
      BUX=VM1/VSHELL
      BUY=VM2/VSHELL
      BUZ=VM3/VSHELL
      IF(IS1.EQ.1) GO TO 301
      PRINT 999,COUL5,COUL6,COUL7,CUUL5,CUUL6,CUUL7,C1,S1
      PRINT 999,CARL5,CARL6,CARL7,VM1,VM2,VM3,XAI,YAI,ZAI
      PRINT 999,XAIM,YAIM,ZAIM,VSHELL,BUX,BUY,BUZ
  301 CONTINUE
      TSTAB=100.
      MPEN=0
      DO 444 M=1,NELIPS
      CAY1=(XAIM-XDH(M))/BDHX(M)
      CAY3=(YAIM-YDH(M))/BDHY(M)
      CAY5=(ZAIM-ZDH(M))/BDHZ(M)
      CAY2=CARL5/BDHX(M)
      CAY4=CARL6/BDHY(M)
      CAY6=CARL7/BDHZ(M)
      CAY7=CAY2**2+CAY4**2+CAY6**2
      CAY8=2.*(CAY1*CAY2+CAY3*CAY4+CAY5*CAY6)
      CAY9=CAY1**2+CAY3**2+CAY5**2-1.
      TEMP1=(CAY8*0.5)/CAY7
      TEMP2=CAY9/CAY7
      DISCRI=TEMP1**2-TEMP2
      IF(DISCRI)444,445,445
  445 TTRIAL=-TEMP1-SQRT(DISCRI)
```

```
      IF(TTRIAL-TSTAB)446,444,444
  446 TSTAB=TTRIAL
      MPEN=M
  444 CONTINUE
      DO 460 M=1,4
  460 P(M)=0.
      IF(MPEN)461,461,462
  461 P(1)=1.
      GO TO 500
  462 IF(KILL(MPEN))471,471,472
  471 P(2)=1.
      GO TO 500
  472 TTT=TSTAB
      M=MPEN
      XS=XAIM+CARL5*TTT
      YS=YAIM+CARL6*TTT
      ZS=ZAIM+CARL7*TTT
      AN=2.*(XS-XDH(M))/(BDHX(M)**2)
      BN=2.*(YS-YDH(M))/(BDHY(M)**2)
      CN=2.*(ZS-ZDH(M))/(BDHZ(M)**2)
      CSANG=(-AN*BUX-BN*BUY-CN*BUZ)/SQRT(AN**2+BN**2+CN**2)
      IF(CSANG-.174)481,481,482
  481 P(3)=1.
      GO TO 500
  482 P(4)=1.
  500 CONTINUE
      PRINT 101, P(1),P(2),P(3),P(4)
      SPK=SPK+P(4)
      SPHG=SPHG+P(3)
      SPHS=SPHS+P(2)
      SPM=SPM+P(1)
  101 FORMAT(1H0,7H       =,F6.4,3X,13H              =,F6.4,3X,16H
     *       =,F6.4,3X,7H       =,F6.4)
  999 FORMAT(10(1X,E11.4))
      RETURN
      END
```

```
      SUBROUTINE QCLIM
      COMMON/COMP/IS1,I99,I88,CLMS(10),CLS1(10),CLS2(10),NSAM,Q1,Q2,Q3,
     *Q4,Q5,Q6,W1,W2,W3,W4,W5,W6,IFT,IPLANE,ICON,S2P1,S1P1,A2P1,A1P1,
     *S2P2,S1P2,A2P2,A1P2,S2P3,S1P3,A2P3,A1P3,S2P4,S1P4,A2P4,A1P4,S2P5,
     *S1P5,A2P5,A1P5,S2P6,S1P6,A2P6,A1P6
C     CALCULATE CONTROL LIMITS
      S2P1=W1*CLS2(NSAM)
      S1P1=W1*CLS1(NSAM)
      A2P1=Q1+W1*CLMS(NSAM)
      A1P1=Q1-W1*CLMS(NSAM)
      S2P2=W2*CLS2(NSAM)
      S1P2=W2*CLS1(NSAM)
      A2P2=Q2+W2*CLMS(NSAM)
      A1P2=Q2-W2*CLMS(NSAM)
      S2P3=W3*CLS2(NSAM)
      S1P3=W3*CLS1(NSAM)
      A2P3=Q3+W3*CLMS(NSAM)
      A1P3=Q3-W3*CLMS(NSAM)
      S2P4=W4*CLS2(NSAM)
      S1P4=W4*CLS1(NSAM)
      A2P4=Q4+W4*CLMS(NSAM)
      A1P4=Q4-W4*CLMS(NSAM)
      S2P5=W5*CLS2(NSAM)
      S1P5=W5*CLS1(NSAM)
      A2P5=Q5+W5*CLMS(NSAM)
      A1P5=Q5-W5*CLMS(NSAM)
      S2P6=W6*CLS2(NSAM)
      S1P6=W6*CLS1(NSAM)
      A2P6=Q6+W6*CLMS(NSAM)
      A1P6=Q6-W6*CLMS(NSAM)
      RETURN
      END
```

```
      SUBROUTINE GUID
C** THIS SUBROUTINE TRANSFORMS EARTH FIXED TO LAUNCH COORDINATES.
      COMMON /ANG/SINSO,COSSO,SINSCO,COSSCO
      COMMON /COMB/XD,YD,ZD,XDTG,YDTG,ZDTG,XDEF,YDEF,ZDEF,ZZZ
     *,XDDE,YDDE,ZDDE
C***
      XDTG = XD * COSSO * COSSCO + YD*COSSO*SINSCO - ZD*SINSO
      YDTG = -XD*SINSCO + YD*COSSCO
      ZDTG = XD*SINSO*COSSCO + YD*SINSO*SINSCO + ZD*COSSO
C***
      RETURN
      END
```

```
      SUBROUTINE KSCALE,RETURNS(Z1)
C***  THIS ROUTINE GIVES THE ESTIMATED TIME AT 1000 FT-TO-GO (T-PRIME)
C***  AND THE VALUE OF GAMMA THAT WILL NOT ALLOW ANALOG OVERLOAD (GAMA).
C***  THEY DETERMINE THE VARIABLE SCALE FACTOR EQUATION. THE RETURNED
C***  VALUES ARE G FOR T-PRIME AND GGG FOR GAMA.
      COMMON/COMA/LEVEL,IPTS,XXS(50),XDTGO,YDTGO,ZDTGO,RLB,COSE,SPO,RI,
     *GAM,EDOT,THETAL,RN,
     *              PPX(50),PPY(50),PPZ(50),TIME(50),TMAS(30),XDTGMS(30)
     *             ,YDTGMS(30),ZDTGMS(30),XMAN(4,50),XMISS(7),NT
     *             ,XCOMP,YCOMP,ZCOMP,TAMA(30),DELTAR(30),VM(30),G,GGG
     *,XDO,YDO,ZDO,DXG,DYG,DZG,S2,S3,S4,S5,XDM(30),YDM(30),ZDM(30),
     *RLBK,SCALEP,F1,F2,F3,G1,G2,G3,XC,YC,ZC,S1,RRR,SR,SPL,CTL,STL,
     *CPL,A1,VTI,XE(30),YE(30),ZE(30),ZALT,NERR,CLA,NPX,NX,CLAA(10),
     *PHO,ARG,AAA,SCALET,TREAL,TMA(30),XTA,YTA,ZTA,SCALEV,QMM(10),QM
     *,SA,CA,VMX(50),VMY(50),VMZ(50)
     *,ECOS(17),HR7EE(17),FI7T(9),FCI77(9)
      COMMON/COMP/IS1,I99,I88,CLMS(10),CLS1(10),CLS2(10),NSAM,Q1,Q2,Q3,
     *Q4,Q5,Q6,W1,W2,W3,W4,W5,W6,IFT,IPLANE,ICON,S2P1,S1P1,A2P1,A1P1,
     *S2P2,S1P2,A2P2,A1P2,S2P3,S1P3,A2P3,A1P3,S2P4,S1P4,A2P4,A1P4,S2P5,
     *S1P5,A2P5,A1P5,S2P6,S1P6,A2P6,A1P6
C
C
C***  BEGIN CONVERGING SEARCH FOR T-PRIME AT 1000 FT-TO-GO.
C
      R=RI
      DG=1.
      AN=0.
      G=0.
   10 G=G+DG
      ARG1=G*SQRT(YDTGO**2+ZDTGO**2)
      ARG2=R+G*XDTGO-1000.
      THETA2=ATAN2(ARG1,ARG2)
      NPX=2
      IF (G.GT.20.) GO TO 20
      NB=28
      CALL INTERP(G,TAMA,DELTAR,NB,NPX,DR,NERR)
      VALUE=R-DR*COS(THETA2)+G*XDTGO-1000.
      A=ABS(VALUE)
      IF (A.LT.20.) GO TO 40
      IF (VALUE.GT.0.) GO TO 10
      AN=AN+1.
      G=G-DG
      DG=.1/AN
      GO TO 10
   20 PRINT 30
   30 FORMAT(10X,35H T-PRIME IS GREATER THAN 20 SECONDS)
      RETURN Z1
   40 CONTINUE
      IF(IS1.NE.1) PRINT 50,G,DR
   50 FORMAT(10X,10H T-PRIME =F10.4,4H SEC,5X,26H DELTAR AT 1000 FT TO G
     *O =F10.2,3H FT)
C
C
C***  BEGIN CONVERGING SEARCH FOR GAMA.
C
      DG=1.
      AN=0.
```

```
      GG=0.
   60 GG=GG+DG
      A=-XDTGO*XDTGO
      CALL INTERP(GG,TAMA,DELTAR,NB,NPX,DR,NERR)
      CALL INTERP(GG,TAMA,VM,NB,NPX,V,NERR)
      B=2.*XDTGO*(20000.-R+DR*COS(THETA2))-20000.*V*COS(THETA2)
      C=(R-DR*COS(THETA2))*(20000.-R+DR*COS(THETA2))
      VALUE=A*GG*GG+B*GG+C
      AA=ABS(VALUE)
      IF(AA.LT.10000.) GO TO 70
      IF (VALUE.GT.0.) GO TO 60
      AN=AN+1.
      GG=GG-DG
      DG=.1/AN
      GO TO 60
   70 CONTINUE
      IF(IS1.NE.1) PRINT 80,GG
   80 FORMAT(10X,25H T AT GAMA MIN OVERLOAD =F10.4,4H SEC)
      IF(IS1.NE.1) PRINT 90,DR
   90 FORMAT(10X,30H DELTAR AT GAMA MIN OVERLOAD =F10.1,3H FT)
      IF(IS1.NE.1) PRINT 95,V
   95 FORMAT(10X,26H VM AT GAMA MIN OVERLOAD =F10.1,7H FT/SEC)
      A1=20000./(DR*COS(THETA2)-R-XDTGO*GG)
      GGG=-(1.+A1)*G/GG
      IF(IS1.NE.1) PRINT 96,GGG
   96 FORMAT(10X,8H GAMA = F10.4)
      A2=0.48/(1.+GGG)
      PRINT 97,A2
   97 FORMAT(10X,31H SMALLEST BIT A/D RESOLUTION = F10.4,2X,3H FT)
      RETURN
      END
```

```
      SUBROUTINE PRIME
      COMMON/COMA/LEVEL,IPTS,XXS(50),XDTGO,YDTGO,ZDTGO,RLB,COSE,SPO,RI,
     *GAM,EDOT,THETAL,RN,
     *              PPX(50),PPY(50),PPZ(50),TIME(50),TMAS(30),XDTGMS(30)
     *             ,YDTGMS(30),ZDTGMS(30),XMAN(4,50),XMISS(7),NT
     *             ,XCOMP,YCOMP,ZCOMP,TAMA(30),DELTAR(30),VM(30),G,GGG
     *,XDO,YDO,ZDO,DXG,DYG,DZG,S2,S3,S4,S5,XDM(30),YDM(30),ZDM(30),
     *RLBK,SCALEP,F1,F2,F3,G1,G2,G3,XC,YC,ZC,S1,RRR,SR,SPL,CTL,STL,
     *CPL,A1,VTI,XE(30),YE(30),ZE(30),ZALT,NERR,CLA,NPX,NX,CLAA(10),
     *PHO,ARG,AAA,SCALET,TREAL,TMA(30),XTA,YTA,ZTA,SCALEV,QMM(10),QM
     *,SA,CA,VMX(50),VMY(50),VMZ(50)
     *,ECOS(17),HR7EE(17),FI7T(9),FCI77(9)
      COMMON/MANEUV/ITT,XDEM(30,45),YDEM(30,45),ZDEM(30,45),
     *              XDTGM(30),YDTGM(30),ZDTGM(30),XPRIME,YPRIME,
     *             ZPRIME,TIMES(50),ITRAJ(60),ITAU(60),III,NADJ
      COMMON /ANG/ SINSO,COSSO,SINSCO,COSSCO
      COMMON/DELT/PPXS(50),PPYS(50),PPZS(50),VMXS(50),VMYS(50),VMZS(50)
      COMMON /COMB/XD,YD,ZD,XDTG,YDTG,ZDTG,XDEF,YDEF,ZDEF,ZZZ
     *,XDDE,YDDE,ZDDE
      COMMON /MU/ COSMU1,SINMU1,COSMU2,SINMU2,COSMU3,SINMU3
      COMMON/MISD/XM(3),YM(3),ZM(3),RM(3),NOPT,ICN
      COMMON/COMP/IS1,I99,I88,CLMS(10),CLS1(10),CLS2(10),NSAM,Q1,Q2,Q3,
     *Q4,Q5,Q6,W1,W2,W3,W4,W5,W6,IFT,IPLANE,ICON,S2P1,S1P1,A2P1,A1P1,
     *S2P2,S1P2,A2P2,A1P2,S2P3,S1P3,A2P3,A1P3,S2P4,S1P4,A2P4,A1P4,S2P5,
     *S1P5,A2P5,A1P5,S2P6,S1P6,A2P6,A1P6
      REAL    MU1,MU2,MU3
      DATA  RTD,DTR/57.2957795,0.01745329/
   50 CONTINUE
      TLAST = TIMES(NADJ)
      TFIRST = TIMES(1)
      TSPAN = TLAST - TFIRST
      THALF = TFIRST + TSPAN / 2.
C*** INTERP RETURNS THESE VALUES FROM TAU ADJUSTED TARGET VELOCITY-TIME
C*** MANEUVER TABLE IN EARTH FIXED COORDINATES.
      CALL INTERP (TLAST,TMA,XDM,NT,NPX,XDEL,NERR)
      CALL INTERP (TFIRST,TMA,XDM,NT,NPX,XDEF,NERR)
      XDDE = (XDEL - XDEF) / TSPAN
      CALL INTERP (TLAST,TMA,YDM,NT,NPX,YDEL,NERR)
      CALL INTERP (TFIRST,TMA,YDM,NT,NPX,YDEF,NERR)
      YDDE = (YDEL - YDEF) / TSPAN
      CALL INTERP (TLAST,TMA,ZDM,NT,NPX,ZDEL,NERR)
      CALL INTERP (TFIRST,TMA,ZDM,NT,NPX,ZDEF,NERR)
      ZDDE=(ZDEL-ZDEF)/TSPAN -32.174
C*** COMPONENTS OF AERODYNAMIC ACCELERATION IN EARTH FIXED COORDINATES
      XD = XDDE
      YD = YDDE
      ZD = ZDDE
C*** TRANSFORM THESE TO LAUNCH COORDINATES
      CALL GUID
C*** COMPONENTS OF AERODYNAMIC ACCELERATION IN LAUNCH COORDINATES
      XDDTG = XDTG
      YDDTG = YDTG
      ZDDTG = ZDTG
C*** DEFINE X- Y- AND ZDTGS
      CALL INTERP (THALF,TMA,XDTGM,NT,NPX,XDTGS,NERR)
C*** INTERP RETURNS THESE VALUES FROM TAU ADJUSTED TARGET VELOCITY-TIME
C*** MANEUVER TABLE IN LAUNCH COORDINATES.
```

```
      CALL INTERP (THALF,TMA,YDTGM,NT,NPX,YDTG5,NERR)
      CALL INTERP (THALF,TMA,ZDTGM,NT,NPX,ZDTG5,NERR)
C*** CALCULATE THE ANGLES USED FOR THE T-MATRIX
   51 IF(ZDTG5.NE.0.0) MU1=ATAN2(YDTG5,ZDTG5)
      IF(ZDTG5.EQ.0.0.AND.YDTG5.EQ.0.0) MU1=0.0
      IF(ZDTG5.EQ.0.0.AND.YDTG5.GT.0.0) MU1=90.*DTR
      IF(ZDTG5.EQ.0.0.AND.YDTG5.LT.0.0) MU1=270.*DTR
   52 IF(XDTG5.GT.0.0) GO TO 53
      IF(XDTG5.LT.0.0) GO TO 54
      MU2=0.0
      GO TO 55
   53 IF(YDTG5.EQ.0.0.AND.ZDTG5.EQ.0.0) MU2=270.*DTR
      ARG=SQRT(YDTG5**2+ZDTG5**2)
      IF(YDTG5.NE.0.0.OR.ZDTG5.NE.0.0) MU2=360.*DTR-ATAN2(XDTG5,ARG)
      GO TO 55
   54 IF(YDTG5.EQ.0.0.AND.ZDTG5.EQ.0.0) MU2=90.*DTR
      ARG=SQRT(YDTG5**2+ZDTG5**2)
      IF(YDTG5.NE.0.0.OR.ZDTG5.NE.0.0) MU2=-ATAN2(XDTG5,ARG)
   55 CONTINUE
C*** DEFINE THE VALUES USED IN THE T-MATRIX
      COSMU1 = COS(MU1)
      SINMU1 = SIN(MU1)
      COSMU2 = COS(MU2)
      SINMU2 = SIN(MU2)
      ARG1=ZDDTG*SINMU1-YDDTG*COSMU1
      ARG2=XDDTG*COSMU2+YDDTG*SINMU1*SINMU2+ZDDTG*COSMU1*SINMU2
      MU3=ATAN2(ARG1,ARG2)
      COSMU3 = COS(MU3)
      SINMU3 = SIN(MU3)
      IF(ICN.NE.1.AND.III.EQ.1) GO TO 70
      IF(IS1.EQ.1) GO TO 70
      PRINT 862,LEVEL,XDDTG,YDDTG,ZDDTG,XDTG5,YDTG5,ZDTG5
   70 CONTINUE
C*** CHANGE RADIANS TO DEGREES FOR PRINT OUT
      MU1  = MU1  * RTD
      MU2  = MU2  * RTD
      MU3  = MU3  * RTD
      IF(IS1.EQ.1) GO TO 90
      IF(ICN.EQ.1.OR.III.NE.1) PRINT 901,MU1,MU2,MU3
   90 CONTINUE
  862 FORMAT(10X,53H MANEUVER WAS IN PROGRESS AT TIME OF DATA COLLECTION
     *./10X,8H LEVEL =,I5,5X,8H XDDTG =,F10.4,5X,8H YDDTG =,F10.4,5X,
     *8H ZDDTG =,F10.4/10X,8H XDTG5 =,F10.4,5X,8H YDTG5 =,F10.4,5X,
     *8H ZDTG5 =,F10.4)
  901 FORMAT(1H0,10X,6H MU1 = F8.2,5X,6H MU2 = F8.2,5X,6H MU3 = F8.2)
      RETURN
      END
```

```
OU.INE MISCOM     74/74   OPT=1                          FTNH 4.2+75075      07/16/7

      SUBROUTINE MISCOM
C
C   THIS SUBROUTINE CALCUL   S THE MISS DISTANCES IN THE TARGET
C   COORDINATE SYSTEM
C
      COMMON/MISD   (3),YM(   ,ZM(3),RM(3),N,ICN
      COMMON/DEL   X(50),Y(50)  Z(50),VMXS(50),VMYS(50),VMZS(50)
      COMMON /MU    OSMU1,SI    1,COSMU2,SINMU2,COSMU3,SINMU3
      COMMON /CO    XD,YD,ZD    TG,YDTG,ZDTG,XDEF,YDEF,ZDEF,ZZZ
     *,XDDE,YDDE,    DE
      COMMON/MANEU./ITT,XD    0,45),YDEM(30,45),ZDEM(30,45),
     *              XDTGM(30),YDTGM(30),ZDTGM(30),XPRIME,YPRIME,
     *              ZPRIME,TIMES(50),ITRAJ(60),ITAU(60),III,NADJ
      COMMON/COMA/LEVEL,IPTS,XXS(50),XDTGO,YDTGO,ZDTGO,RLB,COSE,SPO,RI,
     *GAM,EDOT,THETAL,RN,
     *             PPX(50),PPY(50),PPZ(50),TIME(50),TMAS(30),XDTGMS(30)
     *            ,YDTGMS(30),ZDTGMS(30),XMAN(4,50),XMISS(7),NT
     *            ,XCOMP,YCOMP,ZCOMP,TAMA(30),DELTAR(30),VM(30),G,GGG
     *,XDO,YDO,ZDO,DXG,DYG,DZG,S2,S3,S4,S5,XDM(30),YDM(30),ZDM(30),
     *RLSK,SCALEP,F1,F2,F3,G1,G2,G3,XC,YC,ZC,S1,RRR,SR,SPL,CTL,STL,
     *CPL,A1,VTI,XE(30),YE(30),ZE(30),ZALT,NERR,CLA,NPX,NX,CLAA(10),
     *PHO,ARG,AAA,SCALET,TREAL,TMA(30),XTA,YTA,ZTA,SCALEV,QMM(10),QM
     *,SA,CA,VMX(50),VMY(50),VMZ(50)
     *,ECOS(17),HR7EE(17),FI7T(9),FCI77(9)
      COMMON/GTARG/FXA,FYA,FZA,FXB,FYB,FZB,FXC,FYC,FZC,AT,VT
      COMMON/COMD/COSA,COSB,COSG,YA,ZA
      COMMON/COMP/IS1,I99,I88,CLMS(10),CLS1(10),CLS2(10),NSAM,Q1,Q2,Q3,
     *Q4,Q5,Q6,W1,W2,W3,W4,W5,W6,IFT,IPLANE,ICON,S2P1,S1P1,A2P1,A1P1,
     *S2P2,S1P2,A2P2,A1P2,S2P3,S1P3,A2P3,A1P3,S2P4,S1P4,A2P4,A1P4,S2P5,
     *S1P5,A2P5,A1P5,S2P6,S1P6,A2P6,A1P6
      COMMON/EXTRA/NPTS,DX,XTERM,KKK,AS,SEO,SC,ASEO,R,SCALEL,VVVV,C1,
     *XLOS,YLOS,ZLOS,CO,CX,CY,CLY,CLZ,TRP,XX(50),THE
      DIMENSION  T(50),XT(50),YT(50),ZT(50)
      DTR=0.01745329
      DDX=DX
      II=0
      XN=FLOAT(N)
      SMX=0.
      SMX2=0.
C
C*** NEXT CARD IS TAPE SPECIFIC FOR ITT=1 FOR STATIONARY TARGET
      IF(ITT.EQ.1) GO TO 19
      FX = COSMU2 * COSMU3
      FY=SINMU1*SINMU2*COSMU3-COSMU1*SINMU3
      FZ=SINMU1*SINMU3+COSMU1*SINMU2*COSMU3
      FXA = FX
      FYA = FY
      FZA = FZ
      FX=COSMU2*SINMU3
      FY=COSMU1*COSMU3+SINMU1*SINMU2*SINMU3
      FZ=COSMU1*SINMU2*SINMU3-SINMU1*COSMU3
      FXB = FX
      FYB = FY
      FZB = FZ
      FX=-SINMU2
      FY=SINMU1*COSMU2
      FZ = COSMU1 *COSMU2
```

```
      FXC = FX
      FYC = FY
      FZC = FZ
   19 CONTINUE
      IF((ITT.EQ.1.OR.VVVV.LT.338.0).AND.IS1.NE.1) PRINT 67
      IF((ITT.NE.1.OR.VVVV.GE.338.0).AND.IS1.NE.1) PRINT 66
      DO 1 I=1,N
      DX = X(I)
      DY = Y(I)
      DZ = Z(I)
C***  NEXT CARD IS TAPE SPECIFIC FOR ITT=1 FOR STATIONARY TARGET
      IF(ITT.EQ.1) GO TO 30
      IF(VVVV.LT.338.0) GO TO 30
C     TRANSFORM (MISSILE MINUS TARGET) LAUNCH COORDINATES TO GENERALIZED
C     TARGET COORDINATES
C      REFERENCED TO ORIGIN AT TAILPIPE
      XT(I) = FXA*DX + FYA*DY + FZA*DZ
      YT(I) = FXB*DX + FYB*DY + FZB*DZ
      ZT(I) = FXC*DX + FYC*DY + FZC*DZ
C     CALCULATE TARGET ANGLE OF ATTACK
      AAA=1116.89+0.003894*ZZZ
      VT=SQRT(XDEF**2+YDEF**2+ZDEF**2)
      ARG=0.00003*ZZZ
      PHO=0.00237692*EXP(ARG)
      QM=VT/AAA
      NPX=2
      NX=8
      CALL INTERP (QM,QMM,CLAA,NX,NPX,CLA,NERR)
      WT=18500.0
      ST=248.0
      AT=2.*WT*SQRT(XDDE**2+YDDE**2+ZDDE**2)/(PHO*VT*VT*ST*CLA*32.174)
      IF(IS1.EQ.1) GO TO 5
      IF(ICN.EQ.1.AND.I.EQ.1) PRINT 25,AT
    5 CONTINUE
      AT=AT*DTR
      XS=XT(I)
      YS=YT(I)
      ZS=ZT(I)
C     TRANSFORM GENERALIZED TARGET COORDINATES TO TARGET FIXED COORDINATES
C      REFERENCED TO ORIGIN AT TAILPIPE
      XT(I)=ZS*COS(AT)+XS*SIN(AT)
      YT(I)=-YS
      ZT(I)=XS*COS(AT)-ZS*SIN(AT)
      GO TO 4
C     TRANSFORM MISSILE LAUNCH COORDINATES TO TARGET EARTH FIXED
C      COORDINATES. TARGET IS STATIONARY OR MOVING AT LESS THAN 338 FT/SEC
   30 XT(I)=DX*S2-DY*SPL+DZ*S3
      YT(I)=DX*S4+DY*CPL+DZ*S5
      ZT(I)=-DX*STL+DZ*CTL
    4 CONTINUE
      IF(IS1.NE.1) PRINT 15,XT(I),YT(I),ZT(I)
   15 FORMAT(3(3X,E11.4))
      SMX=SMX+XT(I)
    1 SMX2=SMX2+XT(I)*XT(I)
C
      DO 20 I=1,N
      T(I) = YT(I)
```

```
   20 CONTINUE
C    CALCULATE COEFFICIENTS FOR MISSILE PATH PROJECTIONS IN TARGET FIXED
C    COORDINATES
C     REFERENCED TO ORIGIN AT TAILPIPE
   21 SMT=0.0
      SMXT=0.
      DO 2 I=1,N
      SMT=SMT+T(I)
    2 SMXT=SMXT+XT(I)*T(I)
      D=XN*SMX2-SMX*SMX
      B=(SMT*SMX2-SMX*SMXT)/D
      A=(XN*SMXT-SMT*SMX)/D
C
      II= II + 1
C
   99 GO TO(10,11),II
C    YF-AXIS INTERCEPT AND SLOPE OF PATH PROJECTION IN XF-YF PLANE
C     REFERENCED TO ORIGIN AT TAILPIPE
   10 YM(1)= B
      YA=A
C
  102 DO 3 I=1,N
      T(I) = ZT(I)
    3 CONTINUE
      GO TO 21
C    ZF-AXIS INTERCEPT AND SLOPE OF PATH PROJECTION IN XF-ZF PLANE
C     REFERENCED TO ORIGIN AT TAILPIPE
   11 ZM(1)= B
      ZA=A
C
C    XF-AXIS COORDINATE OF PATH PIERCING POINT IN XF-YF PLANE
C     REFERENCED TO ORIGIN AT TAILPIPE
      XM(2)= -ZM(1)/ZA
C    YF-AXIS COORDINATE OF PATH PIERCING POINT IN XF-YF PLANE
C     REFERENCED TO ORIGIN AT TAILPIPE
      YM(2)= YM(1) + YA*XM(2)
C
C    XF-AXIS COORDINATE OF PATH PIERCING POINT IN XF-ZF PLANE
C     REFERENCED TO ORIGIN AT TAILPIPE
      XM(3)= -YM(1)/YA
C    ZF-AXIS COORDINATE OF PATH PIERCING POINT IN XF-ZF PLANE
C     REFERENCED TO ORIGIN AT TAILPIPE
      ZM(3)= ZM(1) + ZA*XM(3)
C
      XM(1)= 0.
      ZM(2)= 0.
      YM(3)= 0.
C    CALCULATE NEAREST APPROACH TO ORIGIN IN YF-ZF,XF-YF,AND XF-ZF PLANES
C     REFERENCED TO ORIGIN AT TAILPIPE
      DO 155 I=1,3
  155 RM(I)= SQRT(XM(I)*XM(I) + YM(I)*YM(I) + ZM(I)*ZM(I))
      DX=DDX
C
   25 FORMAT(10X,26H TARGET ANGLE OF ATTACK = ,F5.2,3X,4H DEG)
   66 FORMAT(//10X,50H TARGET FIXED COORDINATES OF MISSILE REFERENCED TO
     *           10H TAILPIPE.//)
   67 FORMAT(//10X,43H TARGET EARTH FIXED COORDINATES OF MISSILE.//)
```

```
      RETURN
      END
```

```
      SUBROUTINE MEAN(AXM,AYM,AZM,ARM,NR,SX,SY,SZ,SS)
C
C     THIS SUBROUTINE CALCULATES THE MEAN AND STANDARD DEVIATION
C     OF THE MISS DISTANCES
C
      COMMON/COMP/IS1,I99,I88,CLMS(10),CLS1(10),CLS2(10),NSAM,Q1,Q2,Q3,
     *Q4,Q5,Q6,W1,W2,W3,W4,W5,W6,IFT,IPLANE,ICON,S2P1,S1P1,A2P1,A1P1,
     *S2P2,S1P2,A2P2,A1P2,S2P3,S1P3,A2P3,A1P3,S2P4,S1P4,A2P4,A1P4,S2P5,
     *S1P5,A2P5,A1P5,S2P6,S1P6,A2P6,A1P6
      DIMENSION AXM(20),AYM(20),AZM(20),ARM(20)
      XNR=FLOAT(NR)
      XNN= SQRT(XNR/(XNR-1.))
      J=0
C
    9 SS=0.
      SVS=0.
      J=J+1
      DO  18 I=1,NR
      GO TO (11,12,13,14),J
   11 DUM= AXM(I)
      GO TO 15
   12 DUM= AYM(I)
      GO TO 15
   13 DUM= AZM(I)
      GO TO 15
   14 DUM= ARM(I)
   15 SS= SS+DUM
   18 SVS= SVS + DUM*DUM
C
C    MEAN
C
      SS= SS/XNR
      SVS= SVS/XNR
C
C    STANDARD DEVIATION
C
      DOD=SVS-SS*SS
      DODO=ABS(DOD)
      SVS=SQRT(DODO)*XNN
      GO TO (21,22,23,24),J
   21 SX= SS
      SVX= SVS
      GO TO 9
   22 SY= SS
      SVY= SVS
      GO TO 9
   23 SZ= SS
      SVZ= SVS
      GO TO 9
C
   24 PRINT  34
      PRINT  35, SX,SY,SZ,SS
      PRINT  36, SVX,SVY,SVZ,SVS
      IF(I99.NE.1) GO TO 45
      IF(IPLANE.EQ.0) GO TO 45
      IF(IPLANE.EQ.1) GO TO 40
      IF(IPLANE.EQ.2) GO TO 41
```

```
      IF(IPLANE.EQ.3) GO TO 42
   40 IF(SVY.GT.S2P1.OR.SVY.LT.S1P1) IFT=1
      IF(SVZ.GT.S2P2.OR.SVZ.LT.S1P2) IFT=1
      IF(SY.GT.A2P1.OR.SY.LT.A1P1) IFT=1
      IF(SZ.GT.A2P2.OR.SZ.LT.A1P2) IFT=1
      GO TO 45
   41 IF(SVX.GT.S2P3.OR.SVX.LT.S1P3) IFT=1
      IF(SVY.GT.S2P4.OR.SVY.LT.S1P4) IFT=1
      IF(SX.GT.A2P3.OR.SX.LT.A1P3) IFT=1
      IF(SY.GT.A2P4.OR.SY.LT.A1P4) IFT=1
      GO TO 45
   42 IF(SVX.GT.S2P5.OR.SVX.LT.S1P5) IFT=1
      IF(SVZ.GT.S2P6.OR.SVZ.LT.S1P6) IFT=1
      IF(SX.GT.A2P5.OR.SX.LT.A1P5) IFT=1
      IF(SZ.GT.A2P6.OR.SZ.LT.A1P6) IFT=1
   45 CONTINUE
      RETURN
C
   34 FORMAT(/15X2HXT8X2HYT8X2HZT8X2HRM)
   35 FORMAT(/5X4HMEAN4F10.3)
   36 FORMAT(3X6HST DEV4F10.4)
      END
```

```
      SUBROUTINE INTERP(X,XT,YT,NX,NPX,Y,NERR)
C     DESCRIPTION
C         THIS ROUTINE USES THE LAGRANGE INTERPOLATION FORMULA TO
C         CALCULATE THE VALUE Y=F(X) FROM THE FUNCTION TABLE (YT(I)=
C         F(XT(I)). THE TABLE OF INDEPENDENT VARIABLES MUST BE IN
C         INCREASING ORDER BUT DO NOT HAVE TO BE EVENLY SPACED. IF THE
C         DESIRED ARGUMENT X IS NOT IN THE INDEPENDENT VARIABLE TABLE,
C         THE ROUTINE WILL EXTRAPOLATE USING THE NEAREST POINTS TO THE
C         UNKNOWN VALUE. THE DEGREE OF THE INTERPOLATION FORMULA IS A
C         VARIABLE AND MAY BE SELECTED BY THE USER.
C     INPUT
C         1  X        THE INDEPENDENT VARIABLE X.
C         2  XT       TABLE OF INDEPENDENT X VALUES. MUST BE IN
C                     INCREASING ORDER.
C         3  YT       TABLE OF DEPENDENT Y VALUES. YT(I)=F(XT(I))
C         4  NX       NUMBER OF POINTS IN XT.
C         5  NPX      NUMBER OF POINTS USED IN THE INTERPOLATING FORMULA
C                     WITH XT AS THE INDEPENDENT VARIABLE TABLE. (NPX-1)
C                     IS THE DEGREE OF THE INTERPOLATION FORMULA USED.
C     OUTPUT
C         1  Y        THE INTERPOLATED DEPENDENT VALUE. Y=F(X)
C         2  NERR     NERR=0 IF XT(1) .LE. X .LE. XT(NX)
C                     NERR = 1 IF X .LT. XT(1)
C                     NERR = 2 IF X .GT. XT(NX)
C
      DIMENSION XT(1),YT(1)
      NERR = 0
      NP=NPX
      IF(NX .LT. NP)   NP = NX
      NS = (NX+169)/26
      IH = NP/2
      I = 1
      IF(XT(I) - X)30,20,10
   10 IH = 0
   12 NERR = 1
      GO TO 70
   13 NERR = 2
      GO TO 70
   20 Y = YT(I)
      GO TO 999
   30 I = NX
      IF(XT(I) - X)13,20,50
   50 L = IH + 1
      IS = NS+L
      IF(NX-IS)58,58,52
   52 DO 54    I=IS,NX,NS
      IF(XT(I) - X)54,20,56
   54 L = I
      GO TO 58
   56 L = I - NS
   58 DO 60    I=L,NX
      IF(XT(I) - X)60,20,70
   60 CONTINUE
   70 K = I - IH
      N = K + NP - 1
      Y = 0.0
      IF (N - NX)90,90,80
```

```
     80 N = NX
        K = NX-NP+1
     90 DO 120  J=K,N
        P = 1.0
        DO 110  I=K,N
        IF (I-J)100,110,100
    100 P = P * (X - XT(I)) / (XT(J) - XT(I))
    110 CONTINUE
    120 Y = Y + YT(J)*P
    999 RETURN
        END
```

```
      SUBROUTINE FLIGHT
C
C     ANALOG INPUTS(ADCS)
C        DX,DY,DZ        SCALED XXX,YYY,ZZZ
C        DT              TIME
C        XDOT            X-MISSLE MINUS X-TARGET VELOCITY(FT/SEC)
C        YDOT            Y-MISSLE MINUS Y-TARGET VELOCITY(FT/SEC)
C        ZDOT            Z-MISSLE MINUS Z-TARGET VELOCITY(FT/SEC)
C        XXX             X-MISSLE MINUS X-TARGET POSITION(FT)
C        YYY             Y-MISSLE MINUS Y-TARGET POSITION(FT)
C        ZZZ             Z-MISSLE MINUS Z-TARGET POSITION(FT)
C
C     NOTE, DX,DY,DZ OVERLOAD UNTILL 1000 FT TO GO,HOWEVER, XXX YYY ZZZ
C     SHOULD NEVER OVERLOAD
C
C     ANALOG OUTPUTS(DACS)
C        XDTGO,XCOMP     TARGET VELOCITY X-COMPONENT(LAUNCH COORD.)
C        YDTGO,YCOMP     TARGET VELOCITY Y-COMPONENT(LAUNCH COORD.)
C        ZDTGO,ZCOMP     TARGET VELOCITY Z-COMPONENT(LAUNCH COORD.)
C        RLB             APPARENT PLUME LENGTH/BREADTH RATIO
C        COSE            COS OF ANGLE BETWEEN LOS AND CENTER LINE OF TAR.
C        SPO             SCALED PLUME ROTATION ANGLE
C        RI              INITIAL RANGE
C        GAM             SCALED T-PRIME AT 1000 FT TO GO
C        EDOT            SCALED GAMA/T-PRIME
C        THETAL          INITIAL ELEVATION ANGLE OF TARGET(SCALED)
C        RN              FOR MICOM HYBRID   - A RANDOM NUMBER
C                        FOR IRSS APPLICATION  -IRSS IRIS RATIO NO. 7
C
C     DISCRETE INPUTS
C        IIN             CDC-6600 INPUT SENSE LINES
C                        SENSE LINE 0 = 0000000000000001 =   1
C                        SENSE LINE 1 = 0000000000000010 =   2
C                        SENSE LINE 2 = 0000000000000100 =   4
C                        SENSE LINE 3 = 0000000000001000 =   8
C                        SENSE LINE 4 = 0000000000010000 =  16
C                        SENSE LINE 5 = 0000000000100000 =  32      TR-22
C                        SENSE LINE 6 = 0000000001000000 =  64      TR-21
C                        SENSE LINE 7 = 0000000010000000 = 128
C                        SENSE LINE 8 = 0000000100000000 = 256
C
C     DISCRETE OUTPUTS (NOTE BITS ARE NUMBERED 1-16 FROM RIGHT TO LEFT)
C        IOUT            CDC-6600 OUTPUT CONTROL LINES
C                        CNTRL LINE 4 = 0000000000010000 =  16      TR-33
C                        CNTRL LINE 7 = 0000000010000000 = 128      TR-30
C
C     OTHER PROGRAM VARIABLES
C        NPATH           NPATH=1, WAITING FOR SENSE LINE 6 HIGH
C                        NPATH=2, IN REAL TIME LOOP
C        IWRITE          IF IWRITE = 1, WRITE COMMENTS
C        LEVEL           STATUS OF MANEUVER
C                        =-7 IMPLIES NOT IN REAL TIME
C                        =0 IMPLIES IN REAL TIME
C                        +7 IMPLIES TARGET TRAJ TABLE EXCEEDED
C        WMAN            =0 IMPLIES COLLECT DATA, =7 DONT COLLECT
C        MAN             ARRAY FOR STORING INTERPOLATED VALUES
C        MAX             MAXIMUM POINTS COLLECTED
```

```
C       IPTS            MAXIMUM NUMBER OF DATA POINTS THAT CAN BE
C                       ACTUALLY COLLECTED(SEE NADJ IN SUBROUTINE INIT)
C       IND             INDEX FOR ASSIGNING STORAGE TO XCOMP,YCOMP,ZCOMP
C       INDEX           INDEX OF RANGE TABLE ENTRIES
C       KCK             =-1, TIME.LE.TPRIME. =1, TIME.GT.TPRIME
C       ICR2            COUNTER, INITIALLY LENGTH OF ARRAY TS MINUS ONE
C       NPTS            NUMBER OF POINTS IN RANGE TABLE
C       IDA3            LOGICAL VARIABLE, =-1 IMPLIES NOT YET IN 3-LOOP,
C                       =1 IMPLIES FIRST PASS IN 3-LOOP MADE.
C       LEN             NUMBER OF TRUNKING STATION CONNECTOR PAIRS USED
C       SL5             REAL TIME SIGNAL, STATIC CHECK OK
C       SL4             REAL TIME SIGNAL, ICS RECIVED
C       SL6             REAL TIME SIGNAL, RAMP UP READY
C       SL7             REAL TIME SIGNAL, TERMINATION OF REAL TIME
C
C
C     DADIOS PATCHING REQUIREMENTS (ONE IIDIS AND ONE IODIS)
C
C          TRUNKING                FORTRAN                 AD/4 PATCH
C       V-50 TO W-50      FOR /IDIS2/1,IIDIS      TR00-TR07 AND TR20-TR27
C       V-52 TO W-50      FOR /IDIS2/1,IIDIS      TR40-TR47 AND TR60-TR67
C       V-50 TO W-51      FOR /IDIS2/2,IIDIS      TR00-TR07 AND TR20-TR27
C       V-52 TO W-51      FOR /IDIS2/2,IIDIS      TR40-TR47 AND TR60-TR67
C
C       V-51 TO W-60      FOR /ODIS2/1,IODIS      TR10-TR17 AND TR30-TR37
C       V-53 TO W-60      FOR /ODIS2/1,IODIS      TR50-TR57 AND TR70-TR77
C       V-51 TO W-61      FOR /ODIS2/2,IODIS      TR10-TR17 AND TR30-TR37
C       V-53 TO W-61      FOR /ODIS2/2,IODIS      TR50-TR57 AND TR70-TR77
C
C     THE HIGH ORDER CDC-6600 BIT CORRESPONDS TO TR0X, WHERE X=0,2,4,6
C
C     DADIOS ANALOG PATCHING REQUIREMENTS
C
C        W-03 TO V-06      FOR /*ADC1/49,ADC       TR310-317 AND TR330-337
C        W-00 TO V-00      FOR /*ADC1/ADC          TR010-017 AND TR030-037
C
C        W-13 TO V-07      FOR /*DAC1/49,DAC       TR350-357 AND TR370-377
C        W-10 TO V-01      FOR /*DAC1/DAC          TR050-057 AND TR070-077
C        W-11 TO V-03      FOR /*DAC1/17,DAC       TR150-157 AND TR170-177
C
C     THE FIRST ADC IN EACH GROUP OF 16 CORRESPONDS TO AD/4 TR-10.
C     THE FIRST DAC IN EACH GROUP OF 16 CORRESPONDS TO AD/4 TR-50.
C
C
      INTEGER PIN(8),POUT(8)
      REAL MAN(200),MISSED(7),MISS,LAUNCH
      INTEGER WMAN
      DIMENSION TS(30)
      EQUIVALENCE (TS(1), TMAS(1)),(MAN(1),XMAN(1)),(MISSED(1),XMISS(1))
      COMMON/EEEEE/IT1,KCK,ICR2,IDA3,IND,INDEX,MAX,IWRITE,NPATH,STATUS
      COMMON/COMA/LEVEL,IPTS,XXS(50),XDTGO,YDTGO,ZDTGO,RLB,COSE,SPO,RI,
     *GAM,EDOT,THETAL,RN,
     *             PPX(50),PPY(50),PPZ(50),TIME(50),TMAS(30),XDTGMS(30)
     *            ,YDTGMS(30),ZDTGMS(30),XMAN(4,50),XMISS(7),NT
     *            ,XCOMP,YCOMP,ZCOMP,TAMA(30),DELTAR(30),VM(30),G,GGG
     *,XDO,YDO,ZDO,DXG,DYG,DZG,S2,S3,S4,S5,XDM(30),YDM(30),ZDM(30),
     *RLOK,SCALEP,F1,F2,F3,G1,G2,G3,XC,YC,ZC,S1,RRR,SR,SPL,CTL,STL,
```

```
     *CPL,A1,VTI,XE(30),YE(30),ZE(30),ZALT,NERR,CLA,NPX,NX,CLAA(10),
     *PHO,ARG,AAA,SCALET,TREAL,TMA(30),XTA,YTA,ZTA,SCALEV,QMM(10),QM
     *,SA,CA,VMX(50),VMY(50),VMZ(50)
     *,ECOS(17),HR7EE(17),FI7T(9),FCI77(9)
C
      COMMON/*ADC1/ADIN(10)
      COMMON/*DAC1/LAUNCH(11)
      COMMON/*IDIS2/IIN
      JAM,ON
      COMMON/*ODIS2/IOUT
      JAM,OFF
      JAM,ON                                                          TEST
      COMMON/*DAC1/17,ADINTST(10)                                     TEST
      JAM,OFF                                                         TEST
      NAMELIST/TEST/LEVEL,IPTS,XXS  ,XDTGO,YDTGO,ZDTGO,RLB,COSE,SPO,RI,
     *GAM,EDOT,THETAL,RN,
     *              PPX,PPY,PPZ,TIME,TMAS,XDTGMS
     *             ,YDTGMS,ZDTGMS,XMAN,XMISS,NT
     *             ,XCOMP,YCOMP,ZCOMP,TAMA    ,DELTAR    ,VM    ,G,GGG
     *,XDO,YDO,ZDO,DXG,DYG,DZG,S2,S3,S4,S5,XDM    ,YDM    ,ZDM    ,
     *RLBK,SCALEP,F1,F2,F3,G1,G2,G3,XC,YC,ZC,S1,RRR,SR,SPL,CTL,STL,
     *CPL,A1,VTI,XE    ,YE    ,ZE    ,ZALT,NERR,CLA,NPX,NX,CLAA    ,
     *PHO,ARG,AAA,SCALET,TREAL,TMA    ,XTA,YTA,ZTA,SCALEV,QMM    ,QM
     *,SA,CA,VMX    ,VMY    ,VMZ
     *,ECOS    ,HR7EE    ,FI7T    ,FCI77    ,MAN,MISSED,LAUNCH,TS
     *            ,IT1,KCK,ICR2,IDA3,IND,INDEX,MAX
C
C
C     INITIALIZATION OF FLAGS AND COUNTERS
C
      LEVEL=-7
      IT1=1
      KCK=-1
      ICR2=29
      IDA3=-1
      IND=0
      INDEX=0
      IOUT=0
      IWRITE = 1
      MAX=50
      NPATH=1
C
C     CHECK PATCHING STATUS
C
      LEN=6
      PIN(1)=3RV50
      PIN(2)=3RV51
      PIN(3)=3RV52
      PIN(4)=3RV53
      PIN(5)=3RV07
      PIN(6)=3RV06
      CALL PATSTAT(PIN,POUT,LEN)
      DO 10 I=1,LEN
   10 WRITE(6,8000)PIN(I),POUT(I)
C
C     RESERVE HYBRID EQUIPMENT
C
```

```
      CALL RESERVE(IERR)
      WRITE(6,1000)IERR
      IF(IERR.NE.0)STOP
C
C     JAM ALL CONTROL LINES HIGH FOR CHECK
C
      IOUT=2**16-1
C                                                                    TEST
C     JAM OUT PRE-REAL-TIME TEST FUNCTIONS                           TEST
C                                                                    TEST
      ADINTST(1)=-1.0                                                TEST
      ADINTST(2)=0.0                                                 TEST
      ADINTST(3)=0.0                                                 TEST
      ADINTST(4)=-1.0                                                TEST
      ADINTST(5)=0.0                                                 TEST
      ADINTST(6)=0.0                                                 TEST
      ADINTST(7)=0.0                                                 TEST
      ADINTST(8)=-1.0                                                TEST
      ADINTST(9)=0.0                                                 TEST
      ADINTST(10)=0.0                                                TEST
      PAUSE 1
C
C     JAM ALL CONTROL LINES LOW IN PREPARATION FOR REAL TIME
C
      IOUT=0
C
C     WAIT FOR STATIC CHECK COMPLETE(SENSE LINE 5 = BIT 6)
C
      LARA=1
   40 CALL INTCNVT(LARA,1)
      SL5=IIN.AND.32
      IF(SL5.NE.0)GO TO 50
      PAUSE 2
      GO TO 40
   50 CONTINUE
      WRITE(6,3000)
C
C     PRINT CONTENTS OF COMMON BLOCK COMA
C
      WRITE(6,TEST)
C
C     ENTER REAL TIME
C
      CALL SIMRUN(ISTAT)
      WRITE(6,2000)ISTAT
      CALL REMARK(17H JOB IN REAL TIME)
      CALL BHOLD
C
C     PRINT REMAINDING RCT AND AND REAL TIME ERRORS
C
      TIMELFT=STATUS.A.37777778
      RCTOVER=STATUS.A.400000000000000000B
      ADCOVER=STATUS.A.100000000000000000B
      DACOVER=STATUS.A.200000000000000000B
      ADDOVER=STATUS.A.400000000000000000B
      WRITE(6,9001)STATUS
 9001 FORMAT(*UNITS OF RCT REMAINDING = *,F5.0)
```

```
      IF(RCTOVER.GT.0.0)WRITE(6,9002)
 9002 FORMAT(* RCT OVERUN*)
      IF(ADCOVER.GT.0.0)WRITE(6,9003)
 9003 FORMAT(* OVER VOLTAGE ON ADCS*)
      IF (DACOVER.GT.0.0) WRITE (6,9004)
 9004 FORMAT(* OVER VOLTAGE ON DACS*)
      IF(ADDOVER.GT.0.0)WRITE(6,9005)
 9005 FORMAT(* ADDRESS OUT OF RANGE IN REAL TIME*)
      WRITE(6,TEST)
      RETURN
 1000 FORMAT(*1 RESERVATION ERROR CODE=* ,2X,O20)
 2000 FORMAT(18H REAL TIME STATUS=,O20)
 8000 FORMAT(/*0TRUNK  LINE PATCHING IS *,R3,* TO *,R3)
 3000 FORMAT(22H STATIC CHECK COMPLETE)
      END
```

```
:NE REALT      74/74    OPT=1                          FTNH 4.2+75075      07/16/7

      SUBROUTINE REALT
      REAL MAN(200),MISSED(7),MISS,LAUNCH
      INTEGER WMAN
      DIMENSION TS(30)
      EQUIVALENCE (TS(1), TMAS(1)),(MAN(1),XMAN(1)),(MISSED(1),XMISS(1))
      COMMON/EEEEE/IT1,KCK,ICR2,IDA3,IND,INDEX,MAX,IWRITE,NPATH,STATUS
      COMMON/COMA/LEVEL,IPTS,XXS(50),XDTGO,YDTGO,ZDTGO,RLB,COSE,SPO,RI,
     *GAM,EDOT,THETAL,RN,
     *             PPX(50),PPY(50),PPZ(50),TIME(50),TMAS(30),XDTGMS(30)
     *            ,YDTGMS(30),ZDTGMS(30),XMAN(4,50),XMISS(7),NT
     *            ,XCOMP,YCOMP,ZCOMP,TAMA(30),DELTAR(30),VM(30),G,GGG
     *,XDO,YDO,ZDO,DXG,DYG,DZG,S2,S3,S4,S5,XDM(30),YDM(30),ZDM(30),
     *RLBK,SCALEP,F1,F2,F3,G1,G2,G3,XC,YC,ZC,S1,RRR,SR,SPL,CTL,STL,
     *CPL,A1,VTI,XE(30),YE(30),ZE(30),ZALT,NERR,CLA,NPX,NX,CLAA(10),
     *PHO,ARG,AAA,SCALET,TREAL,TMA(30),XTA,YTA,ZTA,SCALEV,QMM(10),QM
     *,SA,CA,VMX(50),VMY(50),VMZ(50)
     *,ECOS(17),HR7EE(17),FI7T(9),FCI77(9)
      COMMON/*ADC1/ADIN(10)
      COMMON/*DAC1/LAUNCH(11)
      COMMON/*IDIS2/IIN
      JAM,ON
      COMMON/*ODIS2/IOUT
      JAM,OFF
C
      GO TO(1,2),NPATH
C
C     TRANSFER ICS TO AD4
C
    1 CONTINUE
      LAUNCH(1)=XDTGO *.102375
      LAUNCH(2)=YDTGO *.102375
      LAUNCH(3)=ZDTGO *.102375
      LAUNCH(4)=RLB   *.102375
      LAUNCH(5)=COSE  *.102375
      LAUNCH(6)=SPO   *.102375
      LAUNCH(7)=RI    *.102375
      LAUNCH(8)=GAM   *.102375
      LAUNCH(9)=EDOT  *.102375
      LAUNCH(10)=THETAL*.102375
      LAUNCH(11)=RN    *.102375
C
C     SEND STATUS BIT TO AD/4 INDICATING ICS SENT(SENSE LINE 4 = BIT 5)
C
      IOUT=16
C
C     WAIT FOR RAMP UP SIGNAL FROM AD/4(SENSE LINE 6 = BIT 7)
C
      SL6=IIN.AND.64
      IF(SL6.NE.0)NPATH=2
      CALL SIMIDLE
    2 CONTINUE
C
C     READ
C
      WMAN=0
C
C     READ AND SCALE ADCS
```

```
C
      DX=ADIN(1)   /1.02375
      DY=ADIN(2)   /1.02375
      DZ=ADIN(3)   /1.02375
      DT=ADIN(4)   /1.02375
      XDOT=ADIN(5) /1.02375
      YDOT=ADIN(6) /1.02375
      ZDOT=ADIN(7) /1.02375
      XXX=ADIN(8)  /1.02375
      YYY=ADIN(9)  /1.02375
      ZZZ=ADIN(10) /1.02375
C
C     STORE DX,DY,DZ,XDOT,YDOT,AND ZDOT FOR DIAGNOSTIC PURPOSES
C
      DXG=DX*SCALEP
      DYG=DY*SCALEP
      DZG=DZ*SCALEP
      XDO=XDOT
      YDO=YDOT
      ZDO=ZDOT
C
C     COMPARE ACTUAL RANGE WITH RANGE TABLE ENTRY
C
      IF(XXS(INDEX+1).GT.DX)GO TO 5
C
      PPX (INDEX+1)=DX
      PPY (INDEX+1)=DY
      PPZ (INDEX+1)=DZ
      TIME(INDEX+1)=DT
      VMX (INDEX+1)=XDOT
      VMY (INDEX+1)=YDOT
      VMZ (INDEX+1)=ZDOT
C
      WMAN=7
      IDA3=1
C
C     TPRIME
C
    5 CONTINUE
C
      TREAL=DT*SCALET
C
      IF(TREAL.LE.G)GO TO 10
C
C     CKMISS
C
      KCK=1
      SKK=GGG+1.0
      MISS=DX*XDOT+DY*YDOT+DZ*ZDOT
      IF(MISS.LE.0.0)GO TO 10
      MISSED(1)=DX
      MISSED(2)=DY
      MISSED(3)=DZ
      MISSED(4)=XDOT
      MISSED(5)=YDOT
      MISSED(6)=ZDOT
      MISSED(7)=MISS
```

```
 3000 CONTINUE
C
C     SYSTEM HOLD(SENSE LINE 7 = BIT 8)
      IOUT=128
C
C     CHECK REAL TIME STATUS WORD
C
      STATUS=STAT(7777777777777777777B)
C
C     RETURN TO BATCH JOB
C
      CALL SIMSTOP
C
C     CKTIME
C
   10 CONTINUE
      IF(LEVEL.GT.0)GO TO 1000
      IF(DT.LT.TS(1))GO TO 1000
      LEVEL=0
C
C     LOAD
C
   20 CONTINUE
      IF(DT.LE.TS(31-ICR2))GO TO 30
C
C     INC
C
      IT1=IT1+1
      IF(IT1.EQ.NT) GO TO 25
      IF(ICR2.LE.1)GO TO 25
      ICR2=ICR2-1
      GO TO 20
   25 CONTINUE
      LEVEL=7
      GO TO 1000
C
C     INT - INTERPOLATE XCOMP,YCOMP,ZCOMP,XC,YC,ZC
C
   30 CONTINUE
      IT2=IT1+1
      IF(WMAN.EQ.0)GO TO 100
      IND=IND+1
      IF(MAX.LT.IND)GO TO 100
      MAN(4*IND-3)=DT
  100 CONTINUE
C
C     CALC - INTERPOLATE XCOMP,YCOMP,ZCOMP,XC,YC,ZC)
C
      DIV=TS(IT2)-TS(IT1)
      RATIO=(DT-TS(IT1))/DIV
      XCOMP=XDTGMS(IT1)+RATIO*(XDTGMS(IT2)-XDTGMS(IT1))
      XC=XDM(IT1)+RATIO*(XDM(IT2)-XDM(IT1))
      YCOMP=YDTGMS(IT1)+RATIO*(YDTGMS(IT2)-YDTGMS(IT1))
      YC=YDM(IT1)+RATIO*(YDM(IT2)-YDM(IT1))
      ZCOMP=ZDTGMS(IT1)+RATIO*(ZDTGMS(IT2)-ZDTGMS(IT1))
      ZC=ZDM(IT1)+RATIO*(ZDM(IT2)-ZDM(IT1))
      IF(KCK.LT.0)SKK=1.0+GGG*TREAL/G
```

```
C
C     CAT
C
      IF(IDA3.GT.0)GO TO 200
      AAA=0.003894*ZALT+1116.89
      VTI=SQRT(XC*XC+YC*YC+ZC*ZC)
      IF(VTI.LT.338.0)GO TO 200
      ARG=0.00003*ZALT
      PHO=0.00237692*EXP(ARG)
      QM=VTI/AAA
C
C     INTERPOLATE FOR ------------
C
      NPX=2
      NX=8
      CALL INTERP(QM,QMM,CLAA,NX,NPX,CLA,NERR)
      CALL INTERP(TREAL,TMA,XE,NT,NPX,XTA,NERR)
      CALL INTERP(TREAL,TMA,YE,NT,NPX,YTA,NERR)
      CALL INTERP(TREAL,TMA,ZE,NT,NPX,ZTA,NERR)
C
C     CALCULATE COSE,RLB,A1
C
      A1=XTA*XTA+YTA*YTA+ZTA*ZTA
      A1=0.01745329*4.63708424 2*SQRT(A1)/(PHO*VTI*VTI*CLA)
      SA=SIN(A1)
      CA=COS(A1)
      XXX=XXX*20475.0/SKK
      YYY=YYY*20475.0/SKK
      ZZZ=ZZZ*20475.0/SKK
      RRR=XXX*XXX+YYY*YYY+ZZZ*ZZZ
      RFEET=RRR
      RRR=VTI*SQRT(RRR)
      F1=S2*XXX-YYY*SPL+S3*ZZZ
      F2=XXX*S4+YYY*CPL+ZZZ*S5
      F3=CTL*ZZZ-STL*XXX
      SR=SQRT(XC*XC+YC*YC)
      S1=CA+(SA/SR)*ZC
      G1=S1*XC
      G2=S1*YC
      G3=ZC*CA-SA*SR
      E111=(F1*G1+F2*G2+F3*G3)/RRR
      COSE=1.0-E111*E111
      RLB=SQRT(COSE)
      RLB=RLB*RLBK/SCALET
      COSE=E111/1.02375
C
C     COMPUTE IRSS IRIS RATIO NO. 7(PART OF PLUME IRRADIANCE COMP.)
C
      NX=17
      NPX=2
      CALL INTERP(E111,ECOS,HR7EE,NX,NPX,HR7E,NERR)
      HC7FT=1209.675*HR7E*EXP(-2.341*ALOG(RFEET/22965.831))
      P1IRSS=(2.0*ATAN(0.04884004884/RLB))
      P2IRSS=230.34375*RLB/RFEET
      AT7   =((P2IRSS**2)*SIN(P1IRSS))/(2.0*(COS(P1IRSS/2.0))**2)
      PJTU1=0.1*EXP(1.003258*ALOG(AT7))
      FI7=HC7FT/PJTU1
```

```
      NX=9
      NPX=2
      CALL INTERP (FI7,FI7T,FCI77,NX,NPX,FCI7,NERR)
      RN=FCI7
C     WRITE(6,CHECK3)                                                SOFT-
C
C     COMPUTE SPO,THE SCALED PLUME ROTATION ANGLE
C
      RC1=SQRT(F1*F1+F2*F2)
      RC1=F2/RC1
      RCB=ACOS(RC1)
      RCX=-RC1
      RCY=-SIN(RCB)
      IF(F1.GE.0.0)RCY=-RCY
      F11=(F2*G3-F3*G2)/VTI
      F22=(G1*F3-G3*F1)/VTI
      F33=(G2*F1-G1*F2)/VTI
      FCR=SQRT(F11*F11+F22*F22+F33*F33)
      G11=F11*RCX
      G22=F22*RCY
      C111=(G11+G22)/FCR
      T111=ACOS(C111)
      IF(F33.GE.0.0)GO TO 155
      IF(T111.LE.1.570796326)GO TO 156
      TRP=T111-4.71238898
      GO TO 159
  155 TRP=1.570796326-T111
      GO TO 159
  156 TRP=1.570796326+T111
  159 SPO=TRP/SCALET
  200 CONTINUE
C
C     HERE
C
      IF(WMAN.EQ.0)GO TO 2000
      IF(MAX.LE.IND)GO TO 2000
      MAN(4*IND-2)=XCOMP
      MAN(4*IND-1)=YCOMP
      MAN(4*IND)=ZCOMP
      INDEX=INDEX+1
      IF(INDEX.GT.IPTS)GO TO 3000
 2000 CONTINUE
C
C     BYPASS - UPDATE THE DACS
C
      LAUNCH(1)=XCOMP  *.102375
      LAUNCH(2)=YCOMP  *.102375
      LAUNCH(3)=ZCOMP  *.102375
      LAUNCH(4)=RLB    *.102375
      LAUNCH(5)=COSE   *.102375
      LAUNCH(6)=SPO    *.102375
      LAUNCH(7)=RI     *.102375
      LAUNCH(8)=GAM    *.102375
      LAUNCH(9)=EDOT   *.102375
      LAUNCH(10)=THETAL*.102375
      LAUNCH(11)=RN    *.102375
C
```

```
      1000 CALL SIMIDLE
           END
```

APPENDIX G

SUMMARY OF STINGER DIGITAL COMPUTER FUNCTIONS

BOK
DYNAMICS, INC.

PRE-REAL TIME PART

1. Manual Mode Operation

   - Select target crossing angle, altitude, and range from stored table.
   - Select target maneuver from stored table
   - Select target maneuver point at missile launch
   - Adjust maneuver table for selected point at launch
   - Scale maneuver table for analog compatibility
   - Determine type of target from maneuver table
   - Calculate initial target elevation, azimuth, and position
   - Calculate euler angles for earth fixed-to-launch coordinate transformation
   - Transform initial target velocity into launch coordinate system
   - Calculate parameters for the analog variable scale factor equation
   - Generate table of missile range-to-go relative to target for use in controlling the A/D data collection from the analog computer
   - Scale initial launch conditions for analog compatibility
   - Transform target maneuver table into launch coordinates and scale
   - Clear storage area where data is to be accumulated

2. Automatic Mode Operation Has Same Sequence As Manual Mode Except:

   - Target crossing angle, altitude, range, maneuver, and maneuver point are selected from a stored simulation plan
   - There are no manual time delays associated with selection or with problem setup and execution of subsequent simulations stored in the simulation plan

REAL TIME PART

- Convert initial conditions into form compatible with D/A converter
- Wait for AD/4 analog computer to signal readiness to receive initial conditions
- Send initial conditions D/A to the AD/4 computer and start analog real time simulation
- Read A/D relative position, velocity, and time from analog
- Convert A/D information into digital computer notation
- Compare A/D relative position information (range-to-go) with previously generated table
- A/D information is collected and stored as data whenever the above range-to-go comparison is close
- Real time interpolated target maneuver is stored at same time as A/D data collection above
- If A/D information indicates missile has passed the target, the real time part is terminated and control is given to the post-real time part
- Calculate target angle of attack in real time
- Calculate target jet plume apparent length-to-breadth ratio in real time
- Send target velocity components and length-to-breadth ratio D/A to the analog computer
- Collection of last desired A/D data terminates the real time part and gives control to the post real time part

POST-REAL TIME PART

- Scale collected A/D data for digital compatibility
- If nonstationary target, calculate Euler angles for transformation from launch-to-target fixed coordinates
- Calculate the least squares missile approach trajectory from collected A/D data and project miss distance information
- If the target is a high speed jet, perform a lethality test with the probability-of-kill model
- Repeat the simulated flight 20 times and obtain average PK, mean, and standard deviation of miss distance information
- Give control to pre-real time part for next simulation

APPENDIX H

SIMPLIFIED TEST FUNCTIONS

```
      SUBROUTINE DACS
      DIMENSION ADIN(10)
      JAM,ON
      COMMON/*DAC1/17,ADINTST(10)
      JAM,OFF
      COMMON/EEEEE/IT1,KCK,ICR2,IDA3,IND,INDEX,MAX,IWRITE,NPATH
      IF(NPATH.NE.2) TIM=0.0
      IF(NPATH.EQ.2)TIM=TIM+.001
C
C     COMPUTE SIMPLIFIED ADCS
C
      ADIN(1)=-100.0/100.0
      IF(TIM.GE.4.0)ADIN(1)=100.0*(TIM-5.0)/100.0
      ADIN(2)=0.0
      ADIN(3)=0.0
C     NOTE TIME HAS AN ADDED FACTOR OF 10.0
      ADIN(4)=TIM /10.0
      ADIN(5)=(11.0*TIM-5.0)/100.0
      ADIN(6)=(50.0-10.0*TIM)/100.0
      ADIN(7)=(50.0-10.0*TIM)/100.0
      ADIN(8)=(10.0*TIM-50.0)/100.0
      ADIN(9)=0.0
      ADIN(10)=0.0
      DO 10 I=1,10
   10 ADINTST(I)=ADIN(I)
      IF(TIM.GT.15.0) CALL SIMSTOP
      CALL SIMIDLE
      END
```

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.